UCRL-MA-124173

CRETIN User's Manual

Howard A. Scott

Lawrence Livermore National Laboratory

Introduction

CRETIN is a 1-, 2-, and 3-dimensional non-LTE atomic kinetics / radiation transfer code. It self-consistently follows the time evolution of atomic populations and photon distributions as radiation interacts with a low-density plasma. Each element in the plasma is modeled with numerous atomic states, corresponding to the distribution of electrons in various atomic levels. Transitions between levels are caused by interactions between electrons, ions, and photons. Definitions of the atomic levels and information required to calculate transition rates must be supplied by the user, as described in the section on atomic physics input.

This document does not describe the physics modeled by the code or the numerical algorithms used in the code. Much of this information is contained in the paper: "GLF - A Simulation Code for X-Ray Lasers" by H.A. Scott and R.W. Mayle, Applied Physics B, Vol. 58, pp. 35-43, 1994. The description of the treatment of atomic kinetics and radiation transport (for continuum and lines) applies to CRETIN as well as to GLF.

One of CRETIN's main virtues is that it is written in (near-) standard Fortran 77 to be as portable as possible. On some systems, a few routines are written in C to provide capabilities not available from the Fortran compiler. CRETIN can be compiled so as to use only static memory, and in this mode should work with (almost) any compiler. The main problem with this scheme is that running a larger problem (in any sense) may require recompiling a large part of the code. CRETIN can also be compiled to use Cray-style pointers or Fortran 90 pointers, in which case it will allocate the memory it needs dynamically.

NOTE: Fortran 90 pointers are not currently consistent with message passing

CRETIN suffers from a mild case of schizophrenia, as it was originally written (and can still serve) as an astrophysics research tool to model accretion disks. (The name CRETIN arose from a contraction of "accretion".) This document will not discuss this aspect of the code, including relevant generator commands, switch and parameter settings or edits. The schizophrenia should not be apparent to most users.

The latest version of this document should be available at http://www.llnl.gov/def_sci/cretin.

Bug reports, comments and requests concerning CRETIN should be sent to:
hascott@llnl.gov

Acknowledgments

Like most simulation codes, CRETIN owes much to people whose names do not appear on the user's manual title page (and who do not wish to receive bug reports) and to experiences (both positive and negative) with other codes. The treatment of atomic kinetics borrows heavily from that used by Peter Hagelstein and Sam Dalhed in the code XRASER, as does the general approach of evolving temperatures rather than energy densities. Sam Dalhed, Stewart Brown and Ron Mayle have influenced CRETIN's development over an extended period of time. Valuable suggestions, criticisms and/or coding were provided by George Zimmerman, George Maenchen, David Eder, Pieter Dykema and numerous others. For most codes, it is the users who find many of the bugs and drive much of the development, and Alan Wan and Steve Langer have certainly done their part.

Compiling CRETIN

The standard distribution of CRETIN includes a set of source files, sample configuration files and test files. Seven of the source files have the extension *.cmn* and are used as include files during the compilation. There are three sets of these include files. The files in directory *src/main/cmn.cft* are appropriate for compilers which support Cray-style pointers. If your compiler is strictly Fortran 77, then you should use the include files in directory *src/main/cmn.f77*. If your compiler is Fortran 90 and you wish to use Fortran 90 pointers, then use the include files in directory *src/main/cmn.f90*. (Most Fortran 90 compilers on UNIX workstations also support Cray-style pointers). All but one of the remaining source files are Fortran (fixed-format) source files and should have the extensions *.f*, or *.F*. The files with a *.F* extension contain system-dependent coding and must be preprocessed with a C preprocessor before compilation. The file *utilityc.c* is a C source file which is needed for some systems (or for systems using XGRAFIX for runtime graphics).

NOTE: The include file *src/main/cmn.uni/edits.cmn* is for use on systems which do not support Cray pointers to character arrays. In this case, replace *edits.cmn* and define STATIC_CHAR.

On most systems, a makefile is available or can easily be constructed to control compilation. For UNIX systems, a makefile can be constructed by concatenating the three files: *makehead*, *config*, and *maketail*. System-specific information such as compiler flags, libraries and paths resides in *config*. Various sample *config* files are included with the distribution in directory *lib*. For DOS systems, a makefile appropriate for the MKS make utility is also included. For WINDOWS (95/98/NT) systems, additional compilation information is given below. For Macintosh systems, additional compilation information is given below.

NOTE: For systems where make supports VPATH or there is an alternative method of pointing to the include files, they may remain in their directory. On some systems, they will have to be moved or linked to *src/main*.

NOTE: Check the config files for more specific information.

When preprocessing, you must define (using -D) the system that you are using. The following tables lists some systems on which CRETIN has been run and appropriate define flags.

System (OS)	-D flags	*utilityc.c*
Sun (SunOs, Solaris)	UNIX, CFT or F90	
SGI (IRIX)	UNIX, CFT or F90	
IBM (AIX)	AIX, CFT or F90	x
HP (HPUX)	HPUX, CFT	x
DEC (Ultrix, OSF/1)	UNIX, CFT or F90	
Cray (Unicos)	UNICOS, CFT or F90	x
Cray T3D	T3D, CFT	x
Linux w/ Absoft F77	UNIX, CFT, ABSOFT	
Linux w/ Absoft Pro Fortran	UNIX, CFT, SIGNALC, STATIC_CHAR	x
Linux w/ f2c or fort77	LINUX	x
DOS w/ F77L[3]	F77L, F90	
DOS/Windows w/ LF90	F90	
Windows 95/98/NT w/ Visual Fortran	DF, (CFT,UNIX) or F90	
Windows 95/98/NT w/ Fortran Powerstation	FL32, F90	
Macintosh w/ Absoft compiler	MAC, CFT, ABSOFT	
Generic	(none)	

For compiling CRETIN using Visual Studio on WINDOWS (95/98/NT) systems, follow these steps:

1. Create a Project of type QuickWin or Console Application. QuickWin will allow the use of multiple windows, dialog boxes and the SciGraph display library – define WINDOWS in this case.

2. Add all the ".f" and ".F" files in …/src/main to the project except message.f and parallel.F (and divertor.F, if not desired). If you want to include TOTAL, add all the ".f" files in …/src/total as well.

3. Change Project->Settings as follows:

Under Fortran:	
Compatibility:	
add "filenames from command line"	(/fpscomp:filesfromcmd)
Preprocessor:	
add "Use FPP"	(/fpp)
add "Preprocessor symbol to FPP only"	(/nodefine)
Predefined Symbols: e.g. DF WINDOWS …	(/define: "DF WINDOWS …")
INCLUDE paths: add …/src/main/cmn.*xxx*	(/include:"…/src/main/cmn.*xxx*")
(if using TOTAL: add …/src/main/total)	(/include:"…/src/main/total")
If using F90:	
Compilation Diagnostics:	
remove "data alignment"	(/warn:noalignments)
If using PDB:	
External Procedures:	
String Length Argument Passing: After All Args	(/iface:nomixed_str_len_arg)
Under Link->Input:	
Object modules: pdb.lib, pml.lib, score.lib	
Ignore libraries: add libcd.lib	(/nodefaultlib:"libcd.lib")
Additional library path: path to above libraries	(/libpath:" ... ")
If using SCIGRAPH:	
Under Link->Input:	
Object modules: scigraph.lib	
Additional library path: path to above library	(/libpath:" ... ")

For compiling CRETIN using Absoft Fortran under MPW on a Macintosh system, you can either modify the sample makefile (*cretin.make* in directory *lib*) or you can construct your own using the MPW development environment. To build your own makefile, you must first use MPW to construct an initial makefile, then modify that makefile to preprocess the ".F" files. Make sure you are in the …:src:main directory and then follow these steps:

1. Under "Tools", choose Create BuildCommands, with "Output File" MWRE Application and "Output filename" cretin.

2. Under "Source Files" add all the ".f" and ".F" files to the project except dlapack.f, dblas.f, message.f and parallel.F (and divertor.F, if not desired). If you want to include TOTAL, add all the ".f" files in ::total as well.

3. Under "Include Paths" -> "F77 Include Paths" add cmn.cft. If you are including TOTAL, add ::total as well.

4. Under "F77 Options", choose the desired optimization (or debug) settings.

5. Under "Libraries", choose both the UNIX and LAPACK libraries. If using PDB, add pdb.lib, pml.lib, and score.lib as well.

6. The "ACM" button will now open the makefile "cretin.make" in an editor window for further modification. Check that the f77 options include the include paths (-I ':cmn.cft:' –I '::total:') and that the libraries include unixlib.o, lapack77.lib and blas.lib. (Note: using the f77 option '-unix' is not sufficient).

7. For conditional compilation, rename all the ".F" files to have the suffix ".i". In the compilation rules section, for each of these files, change the occurrence of "*file*.F" in the first (dependency) line to "*file*.i". Insert the line
   {cpp} {cpp_opts} *file*.i > *file*.F
   before the second (compilation) line. Define the macro {cpp} to point to your system's C-preprocessor and the macro {cpp_opts} to be the arguments to the preprocessor (e.g. -DMAC -DABSOFT -DCFT -DTOTAL …). You may need to include -C -P among the arguments.

   If your system does not have a C-preprocessor, compiling the file crpp.f (in directory *lib*) with
   f77 crpp.f -o crpp -unix -tool
   will produce an acceptable preprocessor.

8. Build the code with "BuildProgram cretin" (or command-B).

You may also need to set the memory available to the code, either with the info dialog box or with the setprefs tool.

If the preprocessing step fails or the compilation fails with many errors, the problem may lie with the file format, especially if the files were moved from a Unix or Windows machine. MPW includes a tool which can take care of many of these translation problems by typing

translate " " ∂n <oldfile >newfile or

translate " " "" <oldfile >newfile (if the previous form inserts extra lines)

where = Control-j (character code 012 = linefeed) and ∂ = Option-d (character code 266). The makefile also includes the character — = Option-f (character code 304). The script crtrans (in directory *lib*) can help with this task.

The conditional compilation accounts for differences between systems such as timing calls, date and time routines, memory management and handling of execute lines and interrupts. The CFT flag denotes a compiler that supports Cray-style pointers, while F90 denotes a Fortran 90 compiler. On Cray systems, the compiler flag which restricts double precision variables to 64 bits (-d p) should be used.

Depending on your system, you may either compile or link to the linear algebra routines used by CRETIN. The libraries to link to are *libslatec.a*, *liblapack.a*, and *libblas.a*, or equivalents. If any one of these is unavailable, you may compile the appropriate file (*dslatec.f*, *dlapack.f*, or *dblas.a*).

NOTE: If you do compile *dslatec.f*, you must modify the two subroutines with system-dependent data statements (*D1MACH* and *I1MACH*) to match your system. If you do not have access to the library and do not wish to use *dslatec.f*, define NO_SLATEC while compiling *utility.F*. In this case, the code will be unable to perform 2-d diffusion calculations.

The neutral diffusion package is included with the standard distribution but is not included in most of the *config* files. To include this package, define DIVERTOR while compiling the code. Using this package in 2-d requires the SLATEC routines.

Runtime graphics are currently available on UNIX systems using XGRAFIX (version 1.94 or earlier). In this case, define XGRAFIX and link to *libXGC.a*, *libXGF.a*, and *libX11.a*. You must also include *utilityc.c* when compiling. XGRAFIX is available from the Plasma Theory and Simulation Group of U.C. Berkeley (http://langmuir.eecs.berkeley.edu/pub/codes/xgrafix) (Note: the source must be modified to use doubles rather than floats). A modified version of XGRAFIX 1.94 (including bug fixes) is available upon request.

Runtime graphics are also available on DOS systems with the Lahey compilers (F77L3, or LF90) using the utility library (UTIL[3,90]) and the Graphoria library (GRAPH[3,90]). In this case, define LAHEY. On WIN 95/98/NT systems with Visual Fortran or Microsoft Fortran Powerstation, the SCIGRAPH library (included under SAMPLES), can be used – bug fixes are available upon request.

Graphical postprocessing can be accomplished in a number of ways. If PDBLIB is available on your system, define PDB and link with *libpdb.a*, *libpml.a*, and *libscore.a*. CRETIN will then produce PDB files directly readable by ULTRA and PDBVIEW. PDBLIB, ULTRA and PDBVIEW are part of the Portable Application Code Toolkit (PACT) which is available through http://www.llnl.gov/def_sci/pact. Portions of the ascii files produced by CRETIN can also be used as input to most graphics programs. PDBLIB is also necessary for producing restart dumps and other binary dump files.

Several other packages which extend CRETIN's capabilities may not be included in the standard distribution. The TOTAL code provides lineshapes for spectral calculations. The SUPER package provides interactive runtime graphics included with PACT. Other packages may become available in the future. Details are available upon request.

Parameters in *param.cmn*: (* : this parameter is not dynamic)

	MT	# of spatial nodes
	MF, MG	# of continuum groups (these must have the same value)
*	MMZ	maximum # of atomic models
*	METR	# of transitions singled out for editing, or # of line pump commands
*	MXX	# of xfiles
*	MXBC, MBC	# of boundary conditions
	MSP	# of energy groups for spectra
*	MTIME	# of timesteps

The following parameters apply only when using message passing

MTL	# of spatial nodes on a single processor
MPROC	# of processors

The remaining parameters should not be changed!

Parameters in *state.cmn*:

MZ	# of atomic models
MION	# of isoelectronic sequences (in all models)
MLEV	# of atomic levels (in all models)
MTR	# of transitions (in all models)
MATR	# of augis commands (in all models)
MAUT	# of commands in an augis subsection
MSH	# of shells in a "shell" section (maximum over all models)
MHOT	# of additional electron distributions
MSRCZ	# of levels in neutral sequences (for neutral diffusion)

Parameters in *edits.cmn*: (* : this parameter is not dynamic when using STATIC_CHAR)

*	MEDIT	# of edits (edits+tables+plots+displays)
*	MDISP	# of displays
*	MENTRY	# of columns per table or curves per plot (maximum of MMENTRY)
*	MWHEN	# of commands for timed execution
	MSAVETIM	Buffer size for saving time edits
	MSAVESNP	Buffer size for saving snapshot edits
	MSAVEINT	Buffer size for time-integrated edits

Parameters in *lines.cmn*: (* : this parameter is not dynamic)

MLINE	# of lines
MLGRP	# of lines transferred simultaneously
MPAIR	# of line-line interactions
MPRD	# of PRD lines
MBIN	# of bins in a single line
MRPRD	# of r-values in PRD tables
MSPRD	# of s-values in PRD tables
MAPRD	# of a-values in PRD tables

If you are not doing PRD calculations and are using static memory, these last three parameters may be set to 1.

In addition, there is one parameter set within *transfer.F* if using static memory:

MRAY	# of rays (directions) for transfer

There are also a few parameters used for problem generation which are set within *generate.F*:

*	MALIAS	# of aliases
	MPROFILE	# of profile commands
	MSTARK	# of Stark commands
	MKERN	# of kernel commands

Parallel Processing

CRETIN supports two models of parallel processing - a shared memory model (using threads for parallelized loops), and a distributed memory message-passing model (using MPI). Both models may be used simultaneously.

The neutral diffusion package is not currently parallelized with either model.

Distributed Memory

To compile the message-passing version, define MP and link to the appropriate libraries for your system. The files *parallel.F* and *message.f* must be included. Message-passing libraries other than MPI may also be used by replacing the MPI calls in *message.f*.

The atomic kinetics is parallelized by zone. Each processor receives a comparable number of zones, regardless of the distribution of elements within the zones. A crude attempt is made at static load balancing at problem startup, using the relative complexities of the atomic models. This will be replaced in the near future with dynamic load balancing using the actual computational expense of each zone.

There are two options for parallelizing the radiation transfer. The default option replicates the spatial mesh on each processor. With this option, both the continuum transfer and spectral calculations are parallelized over frequencies, with each processor receiving an equal number of frequencies. Since the computational expense is identical for each frequency, no further load balancing is necessary. The line transfer is parallelized by line groups, i.e. groups of lines which are transferred simultaneously due to overlaps or other interactions.

The second option decomposes the mesh into domains, with each processor receiving (at most) one domain. Each processor then handles all frequencies, angles and line groups, while exchanging boundary information with other processors. This option is usually less efficient, but scales up to much larger mesh sizes. This option is chosen by the execute line argument "**domains=***k,l,m*".

The only restriction on the physics performed by the parallel code is that with mesh replication (i.e. the default option), the continuum quantities associated with a line are assumed to have constant values, corresponding to line center, over the entire line profile (equivalent to setting switch(38) $\neq$ 0).

Shared Memory

Compiling the shared-memory version requires use of an OpenMP-compliant compiler or preprocessor (see http://www.openmp.org). For systems which are using the GUIDE preprocessor (part of the KAP/Pro toolset from KAI – see http://www.kai.com), replace f77 (or f90) with guidef77 (or guidef90) in the makefile. Specify the OMP_NUM_THREADS environment variable at run time to set the maximum number of threads.

The atomic kinetics is parallelized by zone. Reasonable load balancing is achieved by giving each processor the most expensive zone remaining to be calculated as the processor finishes the previous zone.

Both the continuum transfer and spectral calculations are parallelized over frequencies, with each processor receiving an equal number of frequencies. Since the computational expense is identical for each frequency, no further load balancing is necessary.

In 1-d, the line transfer is parallelized by line groups, i.e. groups of lines which are transferred simultaneously due to overlaps or other interactions. In 2-d and 3-d, the line transfer is parallelized over directions.

Runtime Files

Input

CRETIN requires an ascii generator file to specify the problem to be run and the edits to be produced. Generator commands are discussed in their own section of this manual.

CRETIN requires an ascii atomic datafile for each atomic model specified in the generator. The format is discussed later in this manual. Data types supported include: *coes, coll excite, coll ionize, phxs, phis, augxs, augis* (including *rec, rad, aug* and *cex*), and *cois*. There can be only one of each of these data sections. CRETIN does NOT check the data file for errors.

CRETIN requires a supplementary data file (xfile) for each **xfile** command in the generator. These files can be either ascii or binary. If you specify an ascii xfile, CRETIN will make a binary version (i.e. a file with the same name and extension *.bin*) which you can use in subsequent runs. The format for the xfile is discussed later in this manual. The xfile can be used either for specifying initial or time-dependent conditions.

When running from a restart dump, the only files necessary are the restart dump and the binary xfiles.

The locations of the atomic datafiles and xfiles can be specified by including the path (relative or absolute) with the filename in the **atoms** or **xfile** commands. If the file is not found in the specified directory, CRETIN will search in the directory specified by the environment variables AFILEDIR (for the atomic datafiles) and XFILEDIR (for the xfiles).

Output

CRETIN can produce a number of different output files. The output files will have the same base name as the generator (unless otherwise requested) with different extensions:

extension	file
.tbl	ascii output, tables
.plt	ascii plots
.ult	binary plots (PDB)
.r*xx*	restart dump (PDB)
.s*xx*	spectral dump (PDB)
.d*xx*	dumpfile (PDB)
.qxx	sensitivity dump (PDB)

where *xx* indicates one of a sequence of two-digit numbers: 00, 01, 02, ...

The ascii output (*.tbl*) file will contain a copy of the generator, warning or error messages from CRETIN, and requested tables. If PDB was defined, the requested plots will by default be contained in a binary (*.ult*) file which can be read directly by ULTRA and PDBVIEW. If PDB was not defined, the plots will be contained in an ascii (*.plt*) file in a format suitable for use with a number of graphics packages. Under DOS or WINDOWS, a postprocessor is available to automate producing plots with EASYPLOT. A file format other than (or in addition to) the default can be chosen by setting switch(11) to the appropriate value.

Restart dumps, spectral dumps, dumpfiles, and sensitivity dumps will be produced upon request, if PDB is available. Each restart dump contains sufficient information from a single time to continue executing the problem. Spectral dumps (sdumps) will contain sufficient information to produce detailed spectra, for one or more times. Dumpfiles

contain the information requested by **dump** commands, for one or more times. Sensitivity dumps contain information sufficient to perform a sensitivity analysis for the zones requested.

WARNING: CRETIN does not check for a previous version before creating most files and will overwrite any such files!

The base name of all output files may be set with the **out=***name* execute line option. This will help avoid overwriting existing files.

Execute Line

The basic form of the execute line which will run the problem specified by generator (or restart) file *problem* is

CRETIN *problem [options]*

There is now a limited capability to modify the problem defined by a restart dump with commands in an additional generator file. The generator can change switch and parameter settings, add new edits and problem specifications. The generator name must appear immediately after the problem name (and must not match any legal option):

CRETIN *problem generator [options]*

Execute line options:

pause	stop in interactive mode after initialization but before the initial timestep
generate	run only through the initial timestep (far enough to produce the initial restart dump)
out=*name*	base all output filenames on *name* (rather than the generator or restart dump name)
size	give the best estimate of what the parameters in *param.cmn*, etc. should be, as well as the memory required to run the problem
threads=*nthreads*	assume the problem will run on *nthreads* threads per processor (only valid with **size**)
procs=*nprocs*	assume the problem will run on *nprocs* processors (only valid with **size**)
domains=*k,l,m*	spatially decompose mesh with *k*, *l*, and *m* domains in the k-, l-, and m-directions (missing arguments have value 1, e.g. **domains**=2 is equivalent to **domains**=2,1,1)
define *var*	assign value 1 to symbol *var* when interpreting generator commands
alias *var value*	assign *value* to symbol *var* when interpreting generator commands
nohold	on Windows and Mac systems, do not keep a window open after execution
help	show execute line options

Other execute line forms:

To identify the current version of CRETIN:

CRETIN version

To produce a binary xfile from the ascii xfile *xname*:

CRETIN *xname* **source**

To produce an ascii xfile from the binary xfile *bname*:

CRETIN *bname* **reconst**

Spectral Postprocessing

CRETIN can be can be run in a manner so as to postprocess a series of restart or spectral dumps to only perform spectral calculations. In this manner, the results of a single simulation can be used to produce spectral information for different code options or edit requests without rerunning the kinetics/transfer simulation. CRETIN will step through each timestep in all restart/spectral dumps with the same base name.

When running in this manner, the generator file must include all definitions of the spectral bins and groups and all edit definitions that are desired. These will not be carried over from the existing dumps. If any **stark** or **kernel** commands are included in the generator, previous definitions concerning these options contained within the dumps will be ignored. If no **stark** or **kernel** commands are included, the previous definitions will be used. Commands dealing with materials, mesh, atomic kinetics, continuum transfer and line transfer are not allowed. Disallowed commands will generate errors during initialization. Parameter settings controlling atomic kinetics, continuum transfer and line transfer will have no effect. If **tstart** and **tquit** are not specified in the generator, the values in the first restart/spectral dump will be used.

For this mode, with first member *name.sxx* (or *name.rxx* for a restart dump) and generator file *generator*, the execute line is

CRETIN *name.sxx generator* **spectra** *[options]*

If the commands in *generator* set switch(92) to <0, the spectral opacities and emissivities will be read from a dump file *name.dxx* with the same base *name* and integer suffix *xx*.

Runtime Control

Under most operating systems, you can suspend execution and interact with CRETIN during a run. Under WINDOWS, hit the right mouse button. Under DOS (with LAHEY defined), hit the escape key. Under most other systems, hit CTRL-C once (two successive CTRL-C's will stop execution). CRETIN will pause at the end of the current timestep. The following commands are available:

help	-	produces this list of options
fin / quit / end	-	finish timestep and quit
kill	-	quit immediately
go [*n*]	-	continue execution [for *n* cycles]
go for Δt	-	continue execution for time Δt
go until *t*	-	continue execution until time *t*
edits	-	make time-independent edits
edits on / off	-	start / stop making time-independent edits
restart	-	make a restart dump
restart on / off	-	start / stop making restart dumps
sdump	-	make a spectral dump
sdump on / off	-	start / stop making spectral dumps
sense on / off / all / *i1:i2:i3 …*	-	start / stop / define zones for sensitivity dumps
tquit	-	print value of quit time
tquit *t*	-	change quit time to *t*
param *i*	-	print value of param *i*
param *i value*	-	change value of param *i* to *value*
switch *i*	-	print value of switch *i*
switch *i value*	-	change value of switch *i* to *value*
size	-	print parameter values and memory usage
timing	-	print cpu usage
trace	-	print trace information during execution
list titles	-	print titles of currently defined edit requests
list *i*	-	print table defined by edit request *i*
list *yvar xvar i1 i2 i3 i4*	-	print table of *yvar* vs. *xvar*
edit i1 i2 i3 i4	-	print value of *edit*
define table / plot / edit / display	-	define a new edit request

If LAHEY or XGRAFIX is defined, the following commands are also available:

plot *i*	-	plot curves defined by edit request *i*
plot *yvar xvar i1 i2 i3 i4*	-	plot *yvar* vs. *xvar*
display *i*	-	plot curves defined by edit request *i*
display *yvar xvar i1 i2 i3 i4*	-	plot *yvar* vs. *xvar*

With LAHEY defined, **plot** will send the requested plot to the screen When a plot appears on the screen, hitting any key will return control to the keyboard. **Display** has the same effect as **plot**, except that CRETIN will continue running and will update the screen at every timestep, sending each defined display to the screen in succession. Hitting the escape key at any point will return control to the keyboard at the end of the current timestep.

With XGRAFIX defined, **plot** and **display** have the same effect. Each will open an additional window with the requested plot. There may be up to MDISP displays defined, including those defined in the generator, subject to the constraint that the total number of edits (**edit**, **table, plot, and display** commands) plus the total number of displays may be no more than MEDIT. The QUIT button has the same effect as **fin**, **quit**, or **end**.

All interactive commands are available for execution at predetermined times with the **when** command except for **define**.

Generator Commands

All commands should be in lower case. Blank lines can be freely interspersed within the generator file. Lines beginning with *c* are interpreted as comments. All input on a line following an exclamation point (!) will be interpreted as a trailing comment. Arguments within brackets are optional. Extra arguments are ignored and can also be used as trailing comments, but may be misinterpreted if the command syntax is extended in the future. A rudimentary interpreter is available through the **alias** command, described in the Miscellaneous section.

Commands may be conditionally included through use of **#ifdef**, **#ifndef**, **#elseif** (or **#elif**), **#else**, and **#endif** clauses. **#ifdef** X will be evaluated as true if X has been given a non-zero numerical value through the use of **#define** or **alias** before the clause is tested. The value may also be set either earlier in the generator file or on the command line.

Materials:

atoms *filename z a*

Specify an atomic model contained in the datafile *filename*. The atomic number and weight are given by *z* and *a*. These will be overwritten if specified with an "atom" command in the datafile. The model will be given an index *iz* corresponding to its order of appearance in the generator file. 'Null' is a valid *filename*.

region *ir1 ir2 tev [tiv [trv]]*

Nodes *ir1* to *ir2* will be initialized with electron temperature *tev*, ion temperature *tiv* and radiation temperature *trv* (in eV). If *tiv (trv)* is absent, *tev* will also be used for the ion (radiation) temperature. The range of nodes *ir1* to *ir2* will apply to all commands from this section (except **atoms**) which do not specify a range of nodes or regions.

regionkl *k1 k2 l1 l2 tev [tiv [trv]]*

Same as **region** but applying to nodes within the 2-d logical region bounded by *(k1,l1)* and *(k2,l2)*

regionklm *k1 k2 l1 l2 m1 m2 tev [tiv [trv]]*

Same as **region** but applying to nodes within the 3-d logical region bounded by *(k1,l1,m1)* and *(k2,l2,m2)*

reg *ireg tev [tiv [trv]]*

Same as **region** but applying to nodes within region *ireg*. The region index *ireg* can be assigned through a **regmap** section in an xfile, a **regmap** command, or is assigned consecutively to each **region**, **regionkl**, or **regionklm** command.

regmult *type value*

Assign region numbers *ireg1, ireg2, ireg3, ireg4, …* to successive nodes in the problem. The node count starts at 1 with the first **regmap** command and continues consecutively through each **regmap** command. This is much the same as the **regmap** section in an xfile, except each line is prefaced with **regmap**.

Type may be "density", "temperature", or “velocity”, or “current”.

regmap *ireg1 ireg2 ireg3 ireg4 …*

Assign region numbers *ireg1, ireg2, ireg3, ireg4, …* to successive nodes in the problem. The node count starts at 1 with the first **regmap** command and continues consecutively through each **regmap** command. This is much the same as the **regmap** section in an xfile, except each line is prefaced with **regmap**.

element *iz ytot [iso1 iso2]*

Assign element *iz* an initial ion population of *ytot* (1/cm^3). Multiple **element** commands can be used in the same region. If *iso1* and *iso2* are present, they specify the range of isoelectronic sequences which will be used for the element *iz* in this region (the default is all isoelectronic sequences). This command cannot be used in the same region with **level** commands.

level *iz iso i y [iso1 iso2]*

Assign the state of element *iz* with (isoelectronic sequence, level) (*iso,i*) an initial ion population of *y* (1/cm^3). If *iso1* and *iso2* are present, they specify the range of isoelectronic sequences which will be used for the element *iz* in this region (the default is all isoelectronic sequences). Multiple **level** commands can be used in the same region. This command cannot be used in the same region with **element** commands.

material *rho ab zb z2b*

Define a background mass density of *rho* (g/cm^3) with average atomic number *ab*, average charge *zb*, and average squared charge *z2b*. This is equivalent to a **background** command, specifying mass density rather than ion density.

background *yb neb ab zb z2b*

Define a background ion density of *yb* (1/cm^3), electron density of *neb* (1/cm^3). The background ions have average atomic number *ab*, average charge *zb*, and average squared charge *z2b*. This is equivalent to a **material** command, specifying ion density rather than mass density..

ne *ne_value*

Set the electron density to *ne_value* (1/cm^3). This will include any background electron density. This value can be used for initialization purposes or can can be fixed for the duration of the run.

bfield *bfield [bx by bz]*

Set the magnetic field value to *bfield* (G). If any of *bx, by, bz* are specified, these will be used as the magnetic field components and the magnitude will be set by the components, i.e. *bfield* will be ignored.

esource *esrce*

Set the free electron heating rate to *esrce* (erg/cm^3/sec).

esourci *esrci*

Set the ion heating rate to *esrci* (erg/cm^3/sec).

scale *type ir1 ir2* $a_0\ a_1\ r_1\ dr_1\ b_1\ a_2\ r_2\ dr_2\ b_2 \ldots$

Scale quantities specified by *type* from nodes *ir1* to *ir2* with position by the factor

$a_0 + \Sigma\, a_i\, [(\text{r-}r_i)/(dr_i)]^{\,b_i}$

Type may be "density", "temperature", or “velocity”, or “current”. This command does not need to be associated with a **region** command.

A type of “density” will scale all mass and number densities. A type of “temperature” will scale electron, ion and radiation temperatuers. This command will be overridden by use of special profiles – see **switch 85**.

scalekl *type k1 k2 l1 l2* $a_0\ a_1\ x_1\ dx_1\ bx_1\ y_1\ dy_1\ by_1\ a_2\ x_2\ dx_2\ bx_2\ y_2\ dy_2\ by_2$ …

Scale quantities specified by *type* for nodes within the 2-d logical region bounded by *(k1,l1)* and *(k2,l2)* by the factor

$a_0 + \Sigma\, a_i\, [(\text{x-}x_i)/(dx_i)]^{\,bx_i}\, [(\text{y-}y_i)/(dy_i)]^{\,by_i}$

Type may be "density" or "temperature". This command does not need to be associated with a **regionkl** command.

The behavior is similar to the **scale** command.

scaleklm *type k1 k2 l1 l2 m1 m2* $a_0\ a_1\ x_1\ dx_1\ bx_1\ y_1\ dy_1\ by_1\ z_1\ dz_1\ bz_1$ …

Scale quantities specified by *type* for nodes within the 3-d logical region bounded by *(k1,l1,m1)* and *(k2,l2,m2)* by the factor

$a_0 + \Sigma\, a_i\, [(\text{x-}x_i)/(dx_i)]^{\,bx_i}\, [(\text{y-}y_i)/(dy_i)]^{\,by_i}\, [(\text{z-}z_i)/(dz_i)]^{\,bz_i}$

Type may be "density" or "temperature". This command does not need to be associated with a **regionklm** command.

The behavior is similar to the **scale** command.

tprofile *[ireg] type time1 time2* $a_0\ a_1\ t_1\ dt_1\ b_1\ a_2\ t_2\ dt_2\ b_2$ …

Scale quantities specified by *type* [for nodes in region *ireg*] with time by the factor

$a_0 + \Sigma\, a_i\, [(\text{time-}t_i)/(dt_i)]^{\,b_i}$

Type may be "density", "temperature", or “velocity”. If no region is specified, the scaling applies to all regions.

Mesh:

If a node-centered mesh is used (switch 42 = 0), the nodes specified in the following commands should cover the entire range (1,nr) or (1,kmax) and (1,lmax). If a zone-centered mesh is used, they should cover the range (2,nr-1) or (2,kmax-1) and (2,lmax-1). CRETIN will add boundary nodes to fill out the ranges. The nodes specified in the **region**, **regionkl** and **regionklm** commands should always cover the entire range.

geometry *type*

Specify geometry for transfer and thermal conduction calculations. Options for *type* include ‘none’ in 0-d,

'plane' (or 'slab'), 'cylinder', 'sphere' and 'wedge' in 1-d, 'xy' and 'rz' in 2-d, and 'xyz' in 3-d. If no **geometry** command is present, the default geometry will be 'plane' if an **rlin**, **rlog**, or **rgeom** command is present, 'xy' if a **quad**, **quadb** or **wedge** command is present, 'xyz' if a **hex** or **cone** command is present, and 'none', otherwise.

rlin *ir1 ir2 rmin rmax*

Linearly interpolate positions between *rmin* at node *ir1* and *rmax* at node *ir2*.

rlog *ir1 ir2 rmin rmax ratio*

Equal ratio zoning between *rmin* at node *ir1* and *rmax* at node *ir2* with ratio *ratio.*

rgeom *ir1 ir2 slope drmin drtot rmin*

Equal ratio zoning between *rmin* at node *ir1* and *rmin+drtot* at node *ir2* with smallest zone size *drmin*. Zone widths increase from *ir1* to *ir2* if *slope* > 0, decrease from *ir1* to *ir2* if *slope* < 0.

quad *k1 k2 l1 l2 xmin xmax ymin ymax ratiok ratiol*

Equal ratio zoning between *(xmin,ymin)* at node *(k1,l1)* and *(xmax,ymax)* at node *(k2,l2)* with ratios *(ratiok,ratiol).*

quadb *k1 k2 l1 l2 x1 y1 x2 y2 x3 y3 x4 y4 ratiok ratiol*

Equal ratio zoning between *(x1,y1)* at node *(k1,l1), (x2,y2)* at node *(k2,l1), (x3,y3)* at node *(k2,l2),* and *(x4,y4)* at node *(k1,l2)* with ratios *(ratiok,ratiol).*

wedge *k1 k2 l1 l2 rmin rmax phimin phimax ratiok ratiol*

Equal ratio zoning of a wedge between *(rmin,phimin)* at node *(k1,l1)* and *(rmax,phimax)* at node *(k2,l2)* with ratios *(ratiok,ratiol).* The angles *phimin* and *phimax* are measured in degrees clockwise from the +y-axis.

hex *k1 k2 l1 l2 m1 m2 xmin xmax ymin ymax zmin zmax ratiok ratiol ratiom*

Equal ratio zoning between *(xmin,ymin,zmin)* at node *(k1,l1,m1)* and *(xmax,ymax,zmax)* at node *(k2,l2,m2)* with ratios *(ratiok,ratiol,ratiom).*

cone *k1 k2 l1 l2 m1 m2 rmin rmax phimin phimax thetamin thetamax ratiok ratiol ratiom*

Equal ratio zoning of a portion of a sphere (not really a cone) between *(rmin,phimin,thetamin)* at node *(k1,l1,m1)* and *(rmax,phimax,thetamax)* at node *(k2,l2,m2)* with ratios *(ratiok,ratiol,ratiom).* The angles *thetamin* and *thetamax* are measured in degrees clockwise from the +z-axis. The angles *phimin* and *phimax* are measured in degrees clockwise from the +y-axis.

Continuum radiation:

frequency *f1 f2 f3 ...*

Define group structure for continuum radiation in terms of frequencies (Hz). The lowest group will extend

from the lowest previously defined group boundary (or zero) to *f1*, the next group will extend from *f1* to *f2*, etc. Transfer will be done at the group centers, evenly spaced between the group boundaries. There may be as many **frequency** commands as necessary to define all the groups.

energy *e1 e2 e3 ...*

Define group structure for continuum radiation in terms of energies (eV). The lowest group will extend from the lowest previously defined group boundary (or zero) to *e1*, the next group will extend from *e1* to *e2*, etc. Transfer will be done at the group centers, evenly spaced between the group boundaries. There may be as many **energy** commands as necessary to define all the groups.

wavelength *w1 w2 w3 ...*

Define group structure for continuum radiation in terms of wavelengths (1/cm). The lowest group will extend from the lowest previously defined group boundary (or zero) to *w1*, the next group will extend from *w1* to *w2*, etc. Transfer will be done at the group centers, evenly spaced between the group boundaries. There may be as many **wavelength** commands as necessary to define all the groups.

fbins *n f1 fn ratio*

Define group structure for continuum radiation in terms of frequencies (Hz). This command will add *n-1* groups (*n* group boundaries) extending from *e1* to *en* with successive group widths increasing by *ratio*. Transfer will be done at the group centers, evenly spaced between the group boundaries. There may be as many **fbins** commands as necessary to define all the groups. These may be combined with **frequency** commands.

ebins *n e1 en [ratio]*

Define group structure for continuum radiation in terms of energies (eV). This command will add *n-1* groups (*n* group boundaries) extending from *e1* to *en* with successive group widths increasing by *ratio*. If *ratio* is absent (or zero), the group boundaries will be logarithmically spaced. Transfer will be done at the group centers, evenly spaced between the group boundaries. There may be as many **ebins** commands as necessary to define all the groups. These may be combined with **energy** commands.

angles *nmu [nphi]*

Number of angles used to set rays for transfer problems. In planar geometry, *nmu* is the number of rays. In cylindrical geometry, *nmu* is the number of angles in the theta direction and rays in the phi direction are spaced no further apart than $\pi/(2\ nphi)$. In spherical geometry, rays are spaced no further apart in mu than 1/*nmu*.

xfilebc *ix mu phi multiplier isotropy*

Assign boundary conditions to the radiation densities (*pbins*) in xfile *ix*. The radiation is assumed to be streaming in the direction (*mu*,*phi*) if *isotropy*=0, or is assumed to be isotropic if *isotropy* is non-zero. The radiation densities in the xfile will be multiplied by *multiplier* for this boundary condition only. The same xfile may be used for multiple boundary conditions

Directions (rays) are defined as follows:

For 1-d geometries, *mu*=cos(θ), where θ is the angle of the ray relative to the normal; *mu*>0 indicates the direction of increasing r, *mu*<0 indicates the direction of decreasing r. For cylindrical geometry, *phi* is the angle

between the normal to the axis and the "x-axis" (chosen so that a ray through the origin has *phi*=0.) and runs from 0. to $\pi/2$.

For xy or rz geometry, *mu*=cos(θ), where θ is the angle between the ray and the z-axis, and *phi* is the angle between the ray and the y-axis, measured clockwise from the +y-axis. For xy geometry, *mu* runs from 0 to 1 and *phi* runs from 0 to 2π. For rz geometry, *mu* runs from -1 to 1 (*mu*>0 indicates the direction of increasing z, *mu*<0 indicates the direction of increasing z) and *phi* runs from $-\pi/2$ to $\pi/2$.

For 3-d (xyz) geometry, *mu*=cos(θ), where θ is the angle between the ray and the +z-axis, and *phi* is the angle between the ray and the y-axis, measured clockwise from the +y-axis. *mu* runs from -1 to 1 and *phi* runs from 0 to 2π.

Spectral radiation:

spectrum *n e1 en [ratio*]

Define spectral group structure for radiation in terms of energies (eV). This command will add *n* energies extending from *e1* to *en* with successive spacings increasing by *ratio*. If *ratio* is absent (or zero), the group boundaries will be logarithmically spaced. Each successive **spectrum** command will effectively add *n*-1 energies, replacing the last previously defined energy with *e1*. There may be as many **spectrum** commands as necessary to define all the spectral groups. The energies defined by successive **spectrum** commands must be monotonically increasing.

The inclusion of one or more **spectrum** commands will result in a detailed spectral calculation being done for each timestep for which edits are requested. If the geometry is 'none', spectral opacities and emissivities will be calculated but not intensities.

spectrum resolution *[tresolve nresolve] [iz] [iso1 iso2 iso3]*

The presence of the **resolution** option on a **spectrum** command results in energies being added to the spectral group structure corresponding to the energy of each photoexcitation (which falls within the limits defined by other **spectrum** commands) from every atomic model. If *tresolve* and *nresolve* are present, energies will be added to resolve a doppler lineshape corresponding to a temperature of *tresolve* (eV), with *nresolve* energies on each side of line center. If *iz* is present, only photoexcitations from atomic model *iz* will be considered. If *iso1, iso2, iso3* are also present, only photoexcitations within the isoelectronic sequence range *iso1-iso3* will be included, while photoexcitations for isoelectronic above *iso2* will be included with *nresolve=0*.

This may eventually be expanded to give the user the capability of specifying the desired resolution of the spectrum in a more meaningful manner.

spectrum lines *[tresolve nresolve] [iz] [iso1 iso2 iso3]*

The presence of the **lines** option on a **spectrum** command results in energies being added to the spectral group structure corresponding to the energy of each photoexcitation singled out with a **line** command (for which the energy falls within the limits defined by other **spectrum** commands). If *tresolve* and *nresolve* are present, energies will be added to resolve a doppler lineshape corresponding to a temperature of *tresolve* (eV), with *nresolve* energies on each side of line center. If both **lines** and **resolution** options are present, the maximum values of *tresolve* and *nresolve* will be used. If *iz* is present, only photoexcitations from atomic model *iz* will be considered. If *iso1, iso2, iso3* are also present, only photoexcitations within the isoelectronic sequence range *iso1-iso3* will be included, while photoexcitations for isoelectronic above *iso2* will be included with *nresolve=0*.

The energy corresponding to a particular line center can be specified in edits with a negative index, where the absolute value of the index gives the line index.

spectral-group *is e1 en*

Define a spectral range (spectral-group) *is* which extends from *e1* to *en* (in eV). Spectral edits can be restricted to a specific energy range by specifying a spectral range index. The full spectrum, defined by all **spectrum** commands, has spectral range index 0. The limits of the spectral range will actually be those energies from the full spectrum which are closest in value to *e1* and *en*.

kernel *iz iso i1 i2*

The **kernel** command specifies a transition for convolving continuum spectra. The local lineshape from the transition connecting levels (*iso*,*i1*) and (*iso*,*i2*) from element *iz*, will be used to convolve the emission and absorption spectra for all bound-free transitions with the same lower level (*iso*,*i1*). For this to work, switch(54) must be set to 2, which will recalculate the continuum spectra on the spectral group structure.

Lineshapes:

stark transition *iz iso i1 i2*
stark manifold *iz iso n1 n2*

Define radiative transitions to be treated with the TOTAL code. If **transition** is specified, the single radiative transition connecting lower state (*iso*,*i1*) with upper state (*iso*,*i2*) in model *iz* will be treated. If **manifold** is specified, all radiative transitions connecting lower states with principal quantum number *n1* with upper states with principal quantum number *n2* within isosequence *iso* in model *iz* will be treated.

Line radiation:

line *iline iz iso1 i1 iso2 i2*

Define a line with index *iline* as the radiative transition connecting lower state (*iso1*,*i1*) with upper state (*iso2*,*i2*) in model *iz*.

linetype *type1 type2 ...*

Define method(s) to use in treating the line defined with the preceding **line** command. Options for *type* include:

	formal	-	formal transfer, Feautrier formalism
+	complin	-	Rybicki linearization
	rybicki	-	Rybicki linearization
	feautrier	-	Feautrier linearization
	formald	-	formal transfer, integral formalism
+	complind	-	linearization, integral formalism
	approximate	-	tridiagonal approximate operator
+	local	-	diagonal approximate operator
	escape	-	escape factor treatment
	none	-	no line transfer treatment
*	nodoppler	-	don't include bulk doppler shifts

	doppler	-	include bulk doppler shifts
*	crd	-	complete redistribution
	prd	-	angle-averaged partial redistribution (RIIA)
	r2a	-	angle-averaged partial redistribution (RIIA)
	r2	-	angle-dependent partial redistribution (RII)
	total	-	lineshape calculated with TOTAL

Default values are marked with a * and may be changed with a **linedefault** command. No **linetype** command is necessary to use the default values. The values marked with a + are possible default values depending on geometry and doppler shifts. In 1-d, the default method is *complin* without doppler shifts and *complind* with doppler shifts. In 2-d and 3-d, only an integral formalism with a local approximate operator (*local*) is available.

linedefault *type1 type2 ...*

Define default method(s) to use in treating lines. Any valid options for **linetype** commands may be specified.

lbins *n1 de1 ratio1 n2 de2 ratio2 ...*

Define the bin structure of the line defined with the preceding **line** command. Starting from line center, there will be *n1* bins spanning an energy *de1* with equal ratio widths using *ratio1*, followed by *n2* bins spanning an energy *de2* with equal ratio widths using *ratio2*, etc. The bin structure will be symmetric about the line center. If this line overlaps with other lines, a combined bin structure will be constructed using the individual line structures. In this case, the total number and positions of the final bins will not match those specified by the **lbins** command. There may be at most one **lbins** command for each line.

resonance *iline1 iline2 de*

Define lines *iline1* and *iline2* to be overlapping with the line center energy of *iline2* greater than that of *iline1* by *de* (in eV). For multiple overlapping lines, a series of **resonance** commands may be used, each with the same *iline1*.

resonant *iline1 iline2 iline3 ...*

Define lines *iline1, iline2, iline3, ...* to be overlapping with line center energies taken from the atomic models.

joinline *iline1 iline2 iline3 ...*

Specify that lines *iline1, iline2, iline3, ...* do not overlap in energy but should be transferred simultaneously.

r2file *filename* [**generate**]

Specify the data file *filename* containing RIIA table to use for PRD calculation. This table is required for either RIIA or RII redistribution. The default is "r2atable". If **generate** is specified, a file of this name will be generated with either specified or default values for Voigt parameters.

rprd *rmax nrprd*

Specify the values of $r = \frac{1}{2}[x-x']$ to use in generating the RIIA table. There will be *nrprd+1* values, evenly spaced from 0. to *rmax*.

sprd *smax nsprd*

Specify the values of $s = \frac{1}{2}[x+x']$ to use in generating the RIIA table. There will be *nsprd+1* values, evenly spaced from 0. to *smax*.

aprd *a1 a2 a3 a4 …*

Specify the values of the Voigt parameters *a1, a2, a3, a4, …* to use in generating the RIIA table. If 0. is not included in the list, it will be added. There may be multiple **aprd** commands.

Edits:

transition *itr iz iso1 i1 iso2 i2*

Define transition index *itr* for edit purposes. All edits having to do with a transition between states (*iso1,i1*) and (*iso2,i2*) in model *iz* will refer the index *itr*. The states must be specified with the lower energy state first and the higher energy state second. For edits which sum or average over states, a value of 0 for *i1* will produce an average over all states in isosequence *iso1*, while a value of 0 for *i2* will produce a sum over all states in isosequence *iso2*.

editray *idir mu phi type*

Define the direction index *idir* for edit purposes. If *type*=0, the direction implied by *idir* will be the direction of that ray which is closest to (*mu, phi*). If *type*<>0, the direction will be that specified by (*mu, phi*).

detector *idtct x y mu phi domega [nmu] [nphi]*
detector *idtct x y mu phi domega* **extent** *dr [nr]*

Define a detector index *idtct* for edit purposes (for 2-d geometries only). The detector is located at position (*x, y*), is pointed in the direction specified by (*mu, phi*) and subtends a solid angle *domega*. If *nmu* and *nphi* are present, the detector solid angle will be discretized with these numbers of angles; otherwise a single direction will be used. If **extent** is specified, the detector is assumed to extend a distance *dr* on either side of the midpoint (*x, y*) in the direction normal to (*mu, phi)* and the detector output is averaged over this distance. In this case, a single direction will be used to discretize the solid angle and *nr* segments will be used to discretize each side of the detector (with a default value of *nr*=1).

table *["title"] [axes]*
plot *["title"] [axes]*
edit *["title"] [axes]*

Define a table/plot with columns/curves to be defined in the following **xvar** and **yvar** commands. If **edit** is specified, both a table and a plot are defined. The plot will be labeled with *title* in the ascii output file but not in the PDB output file (*title* must be in quotes). The numerical value of the final argument *axes* will be interpreted as defining the axis types to be used in interactive displays according to 0: linear-linear, 1: linear-log, 2: log-linear, 3: log-log, other: linear-linear.

display *["title"] [axes]*

Define a display to be sent to the screen with curves to be defined in the following **xvar** and **yvar** commands, labeled with *title*, with axis types defined by *axes* (as above).

send-to-dump

Specify that the current edit (table/plot/edit/display) be put in the dumpfile. Both x-variables and y-variables will appear in the file.

time-integrated [snapshot] *[time1 [time2]]*

Specify that the current edit (table/plot/edit/display) be time-integrated, starting at *time1* (if defined) or problem initialization and ending at *time2* (if defined) or problem termination. If **snapshot** is specified, the time-integrated edit will be output at each edit interval, integrated to the current time, otherwise, the time-integrated edit will be output at problem termination.

time-averaged [snapshot] *[time1 [time2]]*

Specify that the current edit (table/plot/edit/display) be time-averaged, starting at *time1* (if defined) or problem initialization and ending at *time2* (if defined) or problem termination. If **snapshot** is specified, the time-averaged edit will be output at each edit interval, integrated to the current time, otherwise, the time-averaged edit will be output at problem termination.

integrated *[ireg]*
averaged *[ireg]*
mass-averaged *[ireg]*

Specify that the current edit (table/plot/edit/display) be integrated, averaged, or mass-averaged over space. If *ireg* is present, the integration will be over region *ireg*.

rad-units *type1 type2 …*

Specify the units to use for continuum and spectral radiation quantities for the current edit. The default units are c.g.s., i.e. ergs for energies and Hz for frequencies. Available options for *type* are “ergs”, “ev”, and “photons” for the energy unit and “per-hz”, “per-ev”, “per-cm”, and “per-angstrom” for the frequency or wavenumber unit. These units will be used for appropriate y-variables only. The units of the x-variables will remain unchanged.

sp-integrated *[energy1 [energy2]]*

Specify that the current spectral edit (table/plot/edit/display) be integrated over energy/frequency, starting at *energy1* (if defined, in eV) or the first spectral energy and ending at *energy2* (if defined, in eV) or the last spectral energy. The limits of integration will actually be those spectral energy values which are closest in value to *energy1* and *energy2*.

broadened *[resolution]*

Specify that the current spectral edit (table/plot/edit/display) be broadened, i.e. convolved with a gaussian of fwhm (in eV) *resolution* (if defined) or param(82) if *resolution* is not defined.

slice *type index*

For a 3-d problem, restrict the current edit (table/plot/edit/display) be to a single plane in (k,l,m) space, specified by *type*, which may be either “k” or “m” and *index*.

xvar *xedit i1 i2 i3 i4*

Define the x-variable for the plot defined by the preceding **plot** command. *Xedit* is the ascii name of the

edit and must be one of the variables listed below. There may be up to three **xvar** commands for each plot. If any plots specify more than one **xvar** command, those plots can only be viewed with PDBVIEW; all other plots can be viewed with either ULTRA or PDBVIEW. NOTE: For a 2-d problem, "**xvar** r" is expanded to "**xvar** x2d" and "**xvar** y2d". For a 3-d problem, "**xvar** r" is expanded to "**xvar** x3d", "**xvar** y3d" and "**xvar** z3d".

yvar *yedit i1:j1:k1 i2:j2:k2 i3:j3:k3 i4:j4:k4*

Define the y-variable for a curve in the plot defined by the preceding **plot** command. *Yedit* is the ascii name of the edit. There may be multiple **yvar** commands for each plot. Triplet notation *i:j:k* is optional for any index in a **yvar** command.

Independent variables for *xedit*:

cycle, iter, time
ir, r, cdens, x2d, y2d, z2d, x3d, y3d, z3d, xy, k, kr, l, lr, m, mr
ifr, energy, freq, wvl, ebins, fbins, wbins, ifrline, evline, isp, sp_energy, sp_freq, sp_nu, sp_wvl
iso, level, elev

Miscellaneous:

tstart *t*

Start problem at time *t*.

tquit *t*

Stop problem at time *t*.

xfile *filename*

Specify supplementary datafile (xfile) *filename*. The xfile will be given an index *ix* corresponding to its order of appearance in the generator file. 'Null' is a valid *filename*.

xmult *ix type multiplier*

Specify a multiplier to be applied to data of type *type* in xfile ix. *Type* may be (almost) any of the data types allowable in the xfile, i.e. **pbins**, **pbcgs**, **r**, **x2d**, **y2d**, **z2d**, **u**, **u2d**, **v2d**, **w2d**, **durdr**, **d**, **te**, **ti**, **tr**, **bx**, **by**, **bz**, **bfield**, **ne**, **ni, yiso**, **nehot,** or **tehot**. The multipliers for ***x*2d** also are applied to ***x*3d** sections.

boundary *package type ir [***history** *id] multiplier value* or
boundary *package type k1 k2 l1 l2 [***history** *id] value* or
boundary *package type k1 k2 l1 l2 m1 m2 [***history** *id] value*

Specify boundary conditions of a given *type* for a particular physics *package*, which can be "radiation", "conduction", "diffusion", "divertor", "current" or "all". The *type* may be "value", "gradient", "flux", "reflecting", "mirror", "milne" or "recycling". This boundary condition applies at node *ir* (if 1-d), along the line connecting (*k1*,*l1*) and (*k2*,*l2*) (if 2-d), or along the plane specified by *(k1,l1,m1)* and *(k2,l2,m3)* (if 3-d). For 2-d (3-d) geometries, the indices must specify a line of constant *k* or *l* (plane of constant *k*, *l*, or *m*). If **history** is present, time history *id* will be used for the boundary condition, with *value* applied as a multiplier.

The default boundary condition for problem boundaries (nodes, lines or planes) with no **boundary** specifications is:

radiation: free-streaming
conduction: reflecting
diffusion: reflecting
divertor: reflecting
current:

All radiation boundary conditions other than "reflecting" must be specified with an **xfilebc** command.

source *type* …

Specify values for various quantities, depending on *type*. Current valid types are "jbndry", "ni", "yiso", "y", "te", "ti", "tr", "esrce", "esrci", "esrcr", "rho", "ne", "lpump", and "param". Others may be added in the future.

source jbndry *ix* **black-body** *tev [multiplier]*

This form of the **source** command is equivalent to specifying an xfile of index *ix* which includes photon energy densities appropriate for a blackbody of temperature *tev* (eV). The energy densities can be used for boundary conditions as specified by an **xfilebc** command and will be multiplied by *multiplier*, if present.

source ni *ix iz z a [multiplier]*

This form of the **source** command will set the number density of element *iz* to that contained in xfile *iz* if that xfile contains an **ni** section with appropriate identifiers. The identifier in the **znuc** section must match *z*. If an **anuc** section is present, that identifier must match *a*. The number density will be multiplied by *multiplier*, if present. . If $z<0$, the **znuc** and **anuc** sections will be ignored and the mass density in xfile *iz* will be used, and the user should specify a multiplier to correct the values.

source yiso *ix iz iso z a [multiplier]*

This form of the **source** command will set the number density of isosequence *iso* of element *iz* to that contained in xfile *iz* if that xfile contains an **ni** section with appropriate identifiers. The identifier in the **znuc** section must match *z* and the identifier in the **isonuc** section must match *iso*. If an **anuc** section is present, that identifier must match *a*. The number density will be multiplied by *multiplier*, if present.

source y *ix iz iso i z a [multiplier]*

This form of the **source** command will set the number density of level *i* of isosequence *iso* of element *iz* to that contained in xfile *iz* if that xfile contains an **ni** section with appropriate identifiers. The identifier in the **znuc** section must match *z*, the identifier in the **isonuc** section must match *iso*,and the identifier in the **inuc** section must match *i*. If an **anuc** section is present, that identifier must match *a*. The number density will be multiplied by *multiplier*, if present.

NOTE: Use of the **source ni** or **source yiso** form implies that the user is specifying all desired element densities and electron densities, so kinetics quantities will not be updated after calculating populations (equivalent to switch(58) = 1).

source *type [i1 i2 …]* **value/rate history** *id multiplier [ireg/k1 k2 l1 l2 m1 m2]*

This form of the **source** command identifies time history *id* as specifying a quantity *type* to be applied in region *ireg* or into the nodes bounded by *k1 k2 l1 l2 m1 m2*, if either of these is present, or to all regions, if neither specification is present. The values will be interpolated to the current time if **value** is specified, or

will be integrated (and divided by the timestep) to ensure conservation if **rate** is specified.

If *type* is one of “te”, “ti”, “tr”, “rho” or “ne”, this quantity will be set in the same manner as if it were present in an xfile. If *type* is one of “esrce” or “esrci”, it will be treated as an energy source into the free electrons (for “esrce”) or ions (for “esrci”).

If *type* is “param”, it must be followed by a single integer *i1*, and will have the effect of setting the value of param(*i1*) on every timestep.

If *type* is “lpump”, it must be followed by a single integer *iI* corresponding to the source *index* on an **lpump** command. The values specified will be interpreted as photon intensities (in photons/mode) for the radiative transition on the **lpump** command.

history *id [value_multiplier] [time_multiplier] [type p1 p2 p3 …]*

Identify the following **tv** commands as a time history with identifier *id*. All following **tv** commands will have identifier *id* until another **history** command is encountered. The values will be multiplied by *value_multiplier* and the times will be multiplied by *time_multiplier*, if one or both of these values are present. If *type* is present, the values *p1, p2, p3* … will be used as parameters in an analytic form for the time history and no **tv** commands are required.: If **tv** commands are present, *type* will be ignored.

The available analytic forms are:
type = ”gaussian” : value = $\exp(-x^2)$, where x = (time-*p1*) / *p2*.

tv *time value*
tp *time value*

Add a (*time,value*) pair to the time history identified by the previous **history** command. **tp** is an alternative syntax equivalent to **tv.**

restart

Make restart dumps with a frequency controlled by param(87) and switch(65). Restarts may also be made on command in interactive mode.

sdump [spectre] *[filename]*

Make spectral dumps with a frequency controlled by param(91) and switch(72). Spectral dumps may also be made on command in interactive mode. If **spectre** is specified, ascii dumps for the code SPECTRE will be made, otherwise, binary dumps for CRETIN will be made. If *filename* is specified, this will be used as the base for the sequence of spectral dumps, rather than using the generator name.

If CRETIN is running in spectral postprocessing mode from a restart dump and is reading , this command specifies the full filename of the first spectral dump. The default file is the spectral dump made at the time of the restart dump.

dump *variable1 variable2 ...*

Specify variables to output to a PDB file after each timestep for which edits are done. In most cases, the variable is also an edit variable and is described in the edit section. Any combination of the following variables may be specified:

mesh	-	Nodal positions (r if 1-d, (x2d,y2d) if 2-d, (x3d,y3d,z3d) if 3-d)
r	-	Nodal positions

jnu	-	Continuum radiation intensity
absn, emis	-	Continuum absorption, emission
jbar	-	line strengths
radiation	-	jnu + jbar
tev, tiv, trv	-	Electron, ion, radiation temperatures
temperatures	-	tev + tiv + trv
ne, ni	-	Electron, ion number densities
yiso, y	-	iso-sequence, atomic level populations
populations	-	ni + yiso + y
jsp	-	Spectral radiation intensity
kappa_sp, emis_sp	-	Spectral absorption, emission
all	-	all of the above

Edits for which **send-to-dump** have been specified will also appear in the dumpfile.

dumpfile *filename*

Specify the name of the first dumpfile to write or read for the current run to be *filename*. This will be used as the base for the sequence of dumps, rather than using the generator name.

If CRETIN is running in spectral postprocessing mode from a restart or spectral dump reading spectral opacities from a dump file , this command specifies the full filename of the first dump file. The default dump file is the one made at the time of the restart/spectral dump.

sense *i1:j1:k1 i2:j2:k2 …*
sense all

Make sensitivity dumps (PDB) for zones specified by *i1:j1:k1 i2:j2:k2 …*, or for all zones if **all** is specified. Triplet notation (*i:j:k*) is optional. A single family of dumps will hold information for all zones specified. Sensitivity dumps may also be made on command in interactive mode.

nltedump *i1:j1:k1 i2:j2:k2 …*
nltedump all
nltedump rates

Make NLTE workshop dumps (ASCII) for zones specified by *i1:j1:k1 i2:j2:k2 …*, or for all zones if **all** is specified. Triplet notation (*i:j:k*) is optional. If **rates** is specified, rate matrix dumps will also be produced. Files will be named according to workshop specifications.

velocity *ir1 ir2 vel*

Nodes *ir1* to *ir2* will be initialized with 1-d velocity *vel*.

dvdr *ir1 ir2 delv*

Nodes *ir1* to *ir2* will be initialized with 1-d velocity gradient *delv*.

lpump *iz iso1 i1 iso2 i2 value*
lpump *iz iso1 i1 iso2 i2* **source** *index*

Specify photon intensity of *value* (in photons/mode) for the radiative transition between states (*iso1*,*i1*) and (*iso2*,*i2*) in model *iz*. If **source** is specified, the values will be specified by a **source** command which has type “lpump” and identifier *i1=index*.

param *i value*

Set param(*i*) to *value*. All parameters are floating-point numbers.

switch *i value*

Set switch(*i*) to *value*. All switches are integers.

when cycle *i*
when time *t*

This command specifies a time or cycle for executing a series of interactive commands. The commands to be executed follow the **when** command and are terminated by a **done** command. They will be executed at the beginning of cycle *i*, or at the beginning of the first cycle for which the time is greater than or equal to *t*. The interactive commands available for execution are listed in the section on Runtime Control.

done

This command ends a series of interactive commands identified with a **when** command.

alias *word1 word2*
alias *word1 expression*

Substitute *word2* for *word1* for each occurrence in the generator. This is especially useful for setting up zoning and edit requests which can be quickly changed later. Simple arithmetic expressions may be used to calculate a value to be substituted for *word2*. An expression may include numbers, symbols defined with another **alias** command, the operators +, -, *, /, ^ ,**, exp, ln, and log, and parentheses. (Either ^ or ** may be used for exponentiation). The expression is evaluated using algebraic notation and normal operator precedence. All operators must be space delimited. The evaluation of **alias** commands does not depend on order of appearance, but circular definitions are not allowed.

#define *word1 [word2]*

This command is a special form of the **alias** command. If *word2* is not present, it is given the default value of 1. Expressions are not allowed in this command.

Code Options

Atomic Kinetics:

The atomic kinetics LTE or NLTE populations under either steady-state or time-dependent conditions. For LTE, populations are calculated using the Saha equation. For NLTE, populations are calculated with an implicit solution of a first-order (in time) rate equation.

Populations may be initialized for either LTE or steady-state NLTE conditions, or may be initialized explicitly in the generator.

High-density fudges include continuum lowering via a Stewart-Pyatt formula, and multiple knobs for controlling natural line widths.

Relevant switches and parameters:
(see also **switch 33**, **78**, **79** and **80** under Escape Factors)

switch 20 *default value = 1*

Controls choice of LTE / NLTE and radiation used by kinetics:

If switch(20) = 0, calculate LTE populations. In this case, the radiative rates will be evaluated only to calculate the opacities and emissivities. Collisional rates will not be evaluated. This choice is inconsistent with calculating heating rates or heat capacities.

If switch(20) = 1, calculate NLTE populations using rate matrices. The radiation intensities used to calculate the continuum rates will be (in order of priority):
(1) intensities calculated by continuum transfer
(2) intensities from an xfile
(3) zero

If switch(20) = 2, the radiation intensities used to calculate the continuum rates will be Planckian, corresponding to the local electron temperature, T_e

If switch(20) = 3, the radiation intensities used to calculate the continuum rates will be Planckian, corresponding to the local radiation temperature, T_r. The radiation temperature can only be set from an xfile.

switch 25 *default value = 1*

Populations will be calculated assuming steady-state if switch(25) = 0, or assuming time-dependent if switch(25) > 0. No kinetics calculations will be done if switch(25) < 0.

switch 28 *default value = 0*

Controls initialization of populations:

If switch(28) < 0, the initial populations will be LTE for the given electron density (e.g. as fixed by an xfile)

If switch(28) = 0, the initial populations will be LTE for the given ion density (i.e. mass density).

If switch(28) = 1, the initial populations will be steady-state, consistent with radiation for the initial time (either zero intensities, or set from an xfile).

If switch(28) = 2, no initialization calculation will be done. The initial populations must be set from the generator with **level** commands.

switch 32 *default value = 1*

Controls coupling of line intensities to kinetics:

Line intensities will not be seen by any rate calculations (except the line transition) if switch(32) = 0.

Line intensities will be averaged onto the continuum bins and will participate in photoionizations if switch(32) > 0.

Line intensities will be averaged onto the continuum bins and will participate in photoexcitations if switch(32) > 1. The appropriateness of this is left to the user to decide.

switch 55 *default value = 0*

Include continuum lowering if switch(55) ≠ 0.

switch 56 *default value = 2*

If continuum lowering is turned on, the LTE calculations will be iterated a maximum of switch(56) times to converge the populations.

switch 57 *default value = 0*

Controls the calculation of natural line widths for photoexcitations:

If switch(57) >= 0, the natural width will include the lifetimes of both upper and lower states.
If switch(57) < 0, the natural width will include the lifetime of the upper state only.

If switch(57) = 0 or ±1, the natural width will include contributions from all transitions, but will not include the lifetime of a ground state.

If switch(57) = ±2, the natural width will include contributions from all transitions except recombinations, but will not include the lifetime of a ground state.

If switch(57) = ±3, the natural width will include contributions from excitations only, but will not include the lifetime of a ground state.

If switch(57) = ±4, the natural width will include contributions from all transitions.

switch 58 *default value = 0*

Controls updating of material properties (ne, zbar, etc.) after kinetics calculations:

If switch(58) < 0, assume constant electron densities, scale ion densities accordingly and update material properties.

If switch(58) = 0, assume constant ion densities and update material properties.

If switch(58) > 0, do not update material properties – This option is set automatically when ion and electron densities are set from the xfile.

switch 91 *default value = 0*

If switch(91) ≠ 0, make a second pass through the kinetics with updated populations to calculate opacities and emissivities.

switch 102 *default value = 0*

Controls use of level populations read from xfile as specified by **source y** commands:

If switch(102) ≠ 0, those level populations set by the xfile are interpreted as constraints on the kinetics solution. If switch(102) > 0, conservation of the total ion population is enforced as well.

If switch(102) = 0, level populations are calculated normally.

param 56 *default value = 1.00e-2*

Convergence criterion for isosequence populations when iterating timesteps.

param 60 *default value = 1.00e-10*

Isosequences with fractional population greater than param(60) will be considered when testing convergence of populations when iterating timesteps.

param 61 *default value = 0.0*

If the fractional population of an isosequence falls below param(61), that isosequence will not be in the kinetics calculations (for that zone). Judicious use of this parameter can greatly speed up problems where the ionization balance varies widely over time or space, or where the extent of the atomic model is larger than necessary. The window of isosequences which is included can be monitored with the edits **isomin** and **isomax**.

param 62 *default value = 1.00*

When calculating natural line widths, each transition can contribute no more than param(62) times the transition energy.

param 66 *default value = 1.00*

Multiplier for rates from a *coes* section.

param 67 *default value = 1.00*

Multiplier for rates from a *coll excite* section.

param 68 *default value = 1.00*

Multiplier for rates from a *coll ionize* section.

param 69 *default value = 1.00*

Multiplier for rates from a *cois* section.

param 70 *default value = 1.00*

Multiplier for rates from a *phxs* or *phxs1* section.

param 71 *default value = 1.00*

Multiplier for rates from a *phis* section.

param 72 *default value = 1.00*

Multiplier for rates from an *augxs* section.

param 73 *default value = 1.00*

Multiplier for total rates from an *augis* section.

param 74 *default value = 1.00*

Multiplier for rates from a *rec* subsection of an *augis* section.

param 75 *default value = 1.00*

Multiplier for rates from a *rad* subsection of an *augis* section.

param 76 *default value = 1.00*

Multiplier for rates from a *aug* subsection of an *augis* section.

param 77 *default value = 1.00*

Multiplier for rates from a *cex* subsection of an *augis* section.

param 79 *default value = 2.00*

When using continuum lowering, the degeneracy of a state will be decreased by a factor

$\exp(-g^{\text{param(79)}})$

where

$g = [(\Delta E_{max} - \Delta E) / \Delta E_{max}] \exp(\text{param(79)})$

and

ΔE is the energy lowering calculated by the Stewart-Pyatt formula,
ΔE_{max} is the energy lowering necessary for the state to disappear.

param 89 *default value = 1.00*

Multiplier for ΔE calculated by continuum lowering

param 95 *default value = 0.00*

Minimum electron density used to calculated transition rates, expressed as a fraction of the total ion

density for each zone. When dealing with a completely neutral plasma, it can sometimes be helpful to set this to a small number to get things started.

param 96 *default value = 0.50*

Relaxation parameter for electron density iterations. The value used for the next iteration will be

$$n_e^{i+1} = [1\text{-param}(96)]\ n_e^* + \text{param}(96)\ n_e^i$$

where n_e^* is the value obtained from the current kinetics calculation.

Radiation Transfer:

CRETIN separates radiation into two components: continuum and lines. The continuum radiation is treated with formal transfer only, i.e. the radiation intensities are calculated using given, fixed opacities and emissivities. Angular scattering (but not energy scattering) off electrons is included, with the correct angular distribution in 1-d and in an isotropic approximation in 2-d. The line transfer enforces consistency between populations and line strengths through a complete linearization procedure which can handle multiple overlapping and interacting lines. Linearization is applied to individual lines or small groups of lines identified with **resonant**, **resonance**, or **joinline** commands.

In 1-d, Feautrier and integral formalisms are available. Both are applied along long characteristics ("rays") in planar, cylindrical, spherical or "wedge" geometries. In 2-d, only the integral formalism is available, applied along short characteristics for a fixed angle set, i.e. an S_N method.

The following switches and parameters apply to all radiation transfer:

switch 8 *default value = 1*

Controls choice of angle set for radiation transfer:

1-d: use gaussian angles on $[-\pi/2,\pi/2]$ if switch(8) = 0
use double-gaussian angles on $[0,\pi/2]$ if switch(8) $\neq$ 0

2-d: use Carlson A angles if switch(8) = 0
use Carlson B angles if switch(8) = 1
use product ray set based on gaussian angles if switch(8) = 2
use a product ray set based on double-gaussian angles if switch(8) >= 3 or switch(8) < 0

3-d: same as for 2-d

switch 34 *default value = 0*

Controls symmetry boundary conditions:

If 1-d: non-symmetric if switch(34) = 0
symmetric about r=0. if switch(34) $\neq$ 0

If 2-d: ksym = int(switch(34)/10) (10's position)
lsym = mod(switch(34),10) (1's position)
non-symmetric if switch(34) = 0
symmetric about k = 1 if ksym = 1 or 3
symmetric about k = kmax if ksym = 2 or 3
symmetric about l = 1 if lsym = 1 or 3
symmetric about l = lmax if lsym = 2 or 3

If 3-d: ksym = int(switch(34)/100) (100's position)
lsym = int(mod(switch(34),100)/10) (10's position)
msym = mod(switch(34),100) (1's position)
non-symmetric if switch(34) = 0
symmetric about k = 1 if ksym = 1 or 3
symmetric about k = kmax if ksym = 2 or 3
symmetric about l = 1 if lsym = 1 or 3
symmetric about l = lmax if lsym = 2 or 3
symmetric about m = 1 if msym = 1 or 3
symmetric about m = mmax if msym = 2 or 3

For 2-d and 3-d problems, symmetry boundary conditions can also be set with a **boundary** command.

switch 77 *default value =.0*

Don't allow negative continuum opacities if switch(77) ≠ 0.

switch 93 *default value =.0*

Include re-entrant radiation in 2-d and 3-d if switch(93) ≠ 0.

switch 94 *default value =.0*

Use a scratch file for storing 2-d and 3-d ray information if switch(94) ≠ 0.

switch 104 *default value =.0*

Use same memory for continuum, line, and spectrum work spaces if switch(104) ≠ 0.

param 52 *default value = 4.00*

For the integral formalism, the maximum ratio of optical depths for two line segments to be used for a 2nd-order fit. If the ratio exceeds param(52), a 1st-order fit will be used instead.

Continuum Radiation:

Relevant switches and parameters:

switch 17 *default value = 0*

Controls contribution of photoexcitations to opacities and emissivities used in continuum transfer:
don't include any contributions if switch(17) = 0
include contributions from all transitions if switch(17) < 0
include contributions from all transitions not identified as lines if switch(17) > 0.

Contributions from photoexcitations will be integrated over the transition and averaged over the continuum bin containing the transition energy.

switch 35 *default value = 0*

Include free-free opacities and emissivities (bremsstrahlung) if switch(35) = 0, don't include free-free contributions if switch(35) ≠ 0.

switch 36 *default value = 0*

Main control: continuum transfer calculations will be done only if switch(36) ≠ 0, using Feautrier formalism if switch(36) = 1, and using integral formalism, otherwise.

switch 39 *default value = 3*

Maximum number of iterations used to include angular scattering in continuum transfer.

param 5 *default value = 1.00*

Multiplier for electron scattering (Klein-Nishina) cross-sections.

param 28 *default value = 1.00e-2*

Convergence criterion for radiation intensities when iterating continuum transfer.

param 11 *default value = 0.00*

If param(11) ≠ 0.0, use param(11) as the free-free Gaunt factor. This can be useful for matching analytic solutions.

Spectral Radiation:

Spectral radiation includes contributions from both continuum transitions (bound-free and free-free), but from all photoexcitations (bound-bound), as well. The spectral calculation uses a set of energy bins distinct from that used by the atomic kinetics and continuum transfer, and distinct from those used by the line transfer. The spectral bins may be very finely spaced to resolve many transitions without incurring extra expense in the kinetics calculations.

The spectral radiation is treated with a formal transfer only, using populations calculated by the kinetics and augmented by corrections calculated by the line transfer. The results of the spectral calculations are used for edit purposes only, and do not affect the atomic kinetics.

Relevant switches and parameters:
(see also **switch 72**, **param 91** under Timesteps, **param 81**, **82**, **83** under Output, and **switch 86, 87, 88, 90** under Lineshapes)

switch 52 *default value = 0*

Include Stark broadening if switch(52) ≠ 0. Individual lineshapes will still be Voigt profiles, but both the doppler width and natural width will have components due to Stark broadening. This switch does not affect transitions which are treated with TOTAL except for line transfer.

If switch(52) < 0, include Stark broadening for all photoexcitations.
If switch(52) > 0, include Stark broadening only for photoexcitations designated as "lines".

switch 53 *default value = 0*

If switch(53) < -1, do not perform any spectral calculations.

If switch(53) = -1, calculate detailed spectral opacities and emissivities, but do not perform a transfer calculation. The opacities and emissivities can be edited for sent to a dump file.

If switch(53) = 0, use only specified edit rays (and others as necessary) for the transfer calculation.

If switch(53) = 1, use all rays for the transfer calculation.

If switch(53) > 1, use all rays and include coherent electron scattering in the transfer calculation.

switch 54 *default value = 0*

If switch(54) < 0, include only special transitions (i.e. lines and Stark transitions) in the spectral calculation.

If switch(54) = 0, include continuum opacities and emissivities in the spectral calculation, with values obtained by interpolation from the continuum groups.

If switch(54) = 1, include only contributions from photoexcitations.

If switch(54) >= 2, recalculate the continuum opacities and emissivities on the spectral groups.

If switch(54) = 3, convolve the continuum with a Voigt lineshape with a width appropriate for the last discrete Stark-broadened line (Inglis-Teller limit).

If switch(54) = 4, convolve the continuum with a Voigt lineshape with the width calculated for the last discrete line (Inglis-Teller limit) existing in the atomic model.

If switch(54) >= 5, convolve the continuum with a Voigt lineshape with the width calculated for the transition identified with a **kernel** command.

switch 63 *default value = 0*

Include effects of doppler shifts in spectral calculation if switch(63) ≠ 0.

switch 92 *default value = 1*

Calculate spectral opacities and emissivities in chunks if switch(92) = 0, or calculate all frequencies together if switch(92) > 0. Using chunks requires more time, but less memory. If switch(92) < 0, do not calculate this information but read it from existing dump files.

param 78 *default value = 100.*

The contribution of each photoexcitation to the opacity and emissivity will be included out to a minimum of param(78) Lorentz widths (or 6 Doppler widths, if greater) from the transition center.

param 85 *default value = 1.0*

Multiplier for total opacity in spectral calculation.

param 86 *default value = 1.0*

Multiplier for total emissivity in spectral calculation.

param 90 *default value = 0.0 (eV)*

Minimum doppler width assigned to any photoexcitation in spectral calculation. This can be used to artificially broaden transitions to ensure that they will be resolved by the spectral binning.

param 93 *default value = 1.0*

Multiplier for line contributions to opacity and emissivity in spectral calculation (i.e. contributions due to photoexcitations designated as "lines").

Lineshapes:

For the spectral calculation only, selected radiative transitions can be given lineshapes calculated by the TOTAL code, if this was compiled into CRETIN. TOTAL was developed at Universite de Provence at Marseille, France, under Bernard Talin, Roland Stamm and Louis Klein and was modified by Richard Lee. It calculates line profiles using a quasi-static ion microfield approximation and a binary electron impact collision model. A description of the code is given in the user manual, available from Richard Lee.

Relevant switches and parameters:

switch 86 *default value* = 5*0*

Number of microfield values used by TOTAL in lineshape calculations.

switch 87 *default value = 0*

Grouping of transitions for TOTAL calculations will be done individually if switch(87) = 0, by principal quantum number if switch(87) = 1, and by isosequence if switch(87) = 2. Grouping transitions by isosequence requires all levels within that isosequence to have statisitical weights of the form 2(2L+1) or 2J+1.

switch 88 *default value = 0*

Calculate real eigenvalues if switch(88) = 0, calculate complex eigenvalues if switch(88) ≠ 0.

switch 89 *default value = 0*

Include line polarization shifts if switch(89) ≠ 0. These shifts are currently available for Lyman lines only (ref. Nguyen, *et. al.*, Phys. Rev. **A33**, p. 1279, 1986).

Line Radiation:

"Lines"are photoexcitations which have been singled out for special treatment with a **line** command. The exact treatment for each line can be specified by a combination of **linetype** commands and switch and parameter settings. Individual lines are assumed to have a Voigt shape (even for transitions which have been designated for treatment by TOTAL) unless **linetype total** is specified. Both the doppler width and natural width may have components due to Stark broadening.

Relevant switches and parameters:
(see also **switch 57**, **param 62**, **param 81** under Atomic Kinetics and **switch 52** under Spectral Radiation)

switch 37 *default value = 0*

Main control: line transfer calculations will be done only if switch(37) ≠ 0, regardless of any other commands or settings.

switch 38 *default value = 0*

If switch(38) ≠ 0, continuum quantities will be assumed to have constant values over the line profile, corresponding to line center. In this case, only half of a symmetric line will be transferred, if (1) the line is not doppler-shifted, and (2) the line does not overlap any other lines (with non-zero offset of line centers).

switch 40 *default value = 13*

Maximum number of iterations for the approximate operator linearization.

switch 41 *default value = 4*

Number of iterations for the approximate operator before using Ng acceleration.

switch 62 *default value = 0*

If switch(62) ≠ 0, electron scattering will be included as a loss mechanism in the line transfer calculations, under the assumption that the energy shift due to the scattering would result in removal of the photon from the line core. This is a good approximation for isolated, doppler-broadened lines, but may fail under other conditions.

param 50 *default value = 10.*

For complete redistribution (CRD), the tail of the line absorption profile can be analytically integrated to allow for scattering outside of the energy range covered by the line bins, obviating the need to bin optically-thick lines out to many doppler widths. For this purpose, CRD line wings will be assumed to start param(10) doppler widths from line center, or at the last line bin value, whichever is further from line center.

param 51 *default value = 1.00e-3*

Convergence criterion for approximate operator iterations.

param 57 *default value = 1.00e-2*

Convergence criterion for line strengths when iterating timesteps.

param 80 *default value = 1.00e-2*

Minimum optical depth for performing linearization. Lines with a smaller optical depth than param(80) will use formal transfer only, regardless of **linetype** option.

Escape Factors:

Escape factor treatments are available for all lines and photoexcitations. Escape factors will be used under the following conditions:

(1) switch(33) is non-zero and
(2) the transition has not been specified as a "line" and switch(33) > 0, or
the transition has been specified as a "line" with linetype "escape"

The escape factors are non-local, in that an integrated column density of populations is used to calculate the optical depth. If the transition is a "line", the upper and lower level populations are integrated across the mesh. If the transition is not a "line", the iso-sequence population is integrated across the mesh and then scaled by the ratio of the local level populations to the local iso-sequence population. For a 0-d (1-zone) problem, the column density must be set by the user.

The static escape factors (switch(33) = ±1) are available for either planar or spherical geometry. A cylindrical problem will use the planar escape factors.

The Sobolev escape factors (switch(33) = ±2) assume a doppler-broadened line with no continuum opacities. These escape factors are available for planar and cylindrical geometrys. A spherical problem will use the cylindrical escape factors.

The escape factors which interpolate between the static and Sobolev cases (switch(33) = ±3) also assume a doppler-broadened line with no continuum opacities. These escape factors are available only for planar geometry.

The generalized escape factors (switch(33) = ±4) assume a Voigt profile, and also interpolate between the static and Sobolev limits. These escape factors are available for either planar or cylindrical geometry. A spherical problem will use the cylindrical escape factors.

In 2-d geometries, the escape factors are calculated as if the problem were 1-d along l-lines.

Relevant switches and parameters:

switch 33 *default value = 0*

Controls type of escape factor treatment used:
none if switch(33) = 0
static if switch(33) = ±1
Sobolev if switch(33) = ±2
interpolation between static and Sobolev if switch(33) = ±3
generalized escape factor if switch(33) = ±4
none, otherwise

Controls application of escape factors:
apply to all photoexcitations if switch(33) > 0
only apply to lines with linetype "escape" if switch(33) < 0

switch 78 *default value = 0*

Assume a Voigt profile if switch(78) = 0, or a Doppler profile otherwise.

switch 79 *default value = 0*

Include continuum opacities if switch(79) = 0.

switch 80 *default value = 0*

Controls 1-d geometry used for escape factors
problem geometry if switch(80) = 0
planar if switch(80) = 1
cylindrical if switch(80) = 2
spherical if switch(80) = 3

switch 81 *default value = 0*

Use double-sided escape factors if switch(81) = 0, single-sided otherwise.

switch 82 *default value = 0*

Calculate column densities used for escape factors by integrating line populations (for lines) or isoelectronic populations (for other transitions) if switch(82) = 0. If switch(82) > 0, multiply the local density times the distance to boundary. If switch(82) < 0, multiply the local density times the absolute value of param(53).

param 53 *default value = 0.0 (g/cm^2* or *cm^{-2})*

Column density used for escape factor calculations in 0-d problem. If param(53) > 0, the value will be interpreted as a mass column density. If param(53) < 0, the value will be interpreted as a number column density. If switch(82) < 0, the value will be interpreted as a distance to boundary.

param 54 *default value = 1.00e-2*

For (static) optical depths less than param(54), an escape factor of 1.0 will be assigned.

Heating Calculations:

CRETIN can self-consistently calculate the thermal evolution of a plasma, with heating rates determined by atomic processes. The temperature is calculated by implicitly solving a linearized energy balance equation, including electron thermal conduction, and a single timestep can be iterated to assure convergence of the full non-linear equation.

Relevant switches and parameters:
(see also **switch 34** under Radiation Transfer for symmetry boundary conditions)

switch 31 *default value = 0*

Controls type of temperature calculation
none if switch(31) = 0
time-dependent if switch(31) > 0
steady-state if switch(31) < 0

switch 49 *default value = 0*

Control calculation of electron thermal conduction
no thermal conduction if switch(49) = 0
include thermal conduction if switch(49) ≠ 0

If 2-d:
use iccg if switch(49) = ± 1
use ilur if switch(49) = ± 2
use gmres with diagonal preconditioning if switch(49) = ± 3
use gmres with iccg preconditioning if switch(49) = ± 4
use gmres with ilur preconditioning if switch(49) = ± 5
use gmres with no preconditioning otherwise
print convergence diagnostics to screen if switch(49) < 0

switch 50 *default value = 0*

Flux limit the conduction calculations if switch(50) ≠ 0

switch 51 *default value = 0*

If switch(51) ≠ 0, set both the coulomb logarithm and the z-dependent correction term for the conduction coefficient to 1.0. This can be useful for matching analytic solutions.

switch 66 *default value = 100*

The maximum number of iterations used by iccg (if switch(49) = ±1).

switch 67 *default value = 10*

The maximum number of orthogonalizations used by gmres (if switch(49) = ±2 or ±3).

switch 66 *default value = 4*

The maximum number of restarts used by gmres. Set this to -1 for no restarts as 0 will result in the (default) value of 10 restarts.

switch 83 *default value = 0*

Include additional energy sources if switch(83) ≠ 0.

For switch(83) = 1, the energy source is taken to be

$$H = \text{param}(94) \langle Z \rangle [1-(r/r_{max})^2] / T^{1.5} \quad (\text{erg/cm}^3\text{/s})$$

where T is the electron temperature in eV

For switch(83) = 2, the energy source is obtained from a current evolution calculation

switch 84 *default value = 0*

Zone-center the electron conduction coefficients if switch(84) ≠ 0.

param 48 *default value = 1.0*

Multiplier for the electron-ion coupling coefficient

param 55 *default value = 1.0e-2*

Convergence criterion for electron and ion temperatures when iterating timesteps

param 64 *default value = 1.0*

Multiplier for the electron thermal conduction coefficient

param 65 *default value = 1.0*

Multiplier for the thermal conduction flux limiter

param 88 *default value = 1.0e-2*

Convergence criterion for iccg and gmres iterations

param 94 *default value = 0.00*

Coefficient for energy source term – see **switch 83**

Timesteps and convergence:

There are three options for choosing timesteps: (1) the code can match timesteps to those contained in the (first) xfile, (2) the user can set a constant timestep (which can be changed interactively), or (3) timesteps can be chosen automatically by the code, given an initial timestep set by the user. For a steady-state problem, the timestep Δt is irrelevant but must still be provided, if only by accepting default values.

A single timestep can be iterated until the temperatures, iso-sequence populations and line strengths have either converged to user-set criteria or until a maximum number of iterations has been reached.

Relevant switches and parameters:

switch 29 *default value = 2*

Controls type of timesteps
match timesteps to xfile if switch(29) = 0
use constant timesteps (of size Δt = param(41)) if switch(29) = 1
adjust Δt automatically if switch(29) = 2

switch 44 *default value = 0*

Maximum number of iterations for a single timestep.

param 41 *default value = 1.00e-12 (sec)*

Timestep Δt, if using constant timesteps, or initial timestep if using automatic timesteps.

param 42 *default value = 0.20*

Automatic Δt control: maximum change in zbar per timestep.

param 43 *default value = 1.50*

Automatic Δt control: maximum fractional change in Δt per timestep.

param 44 *default value = 1.00e-15 (sec)*

Automatic Δt control: minimum timestep.

param 45 *default value = 1.00 (sec)*

Automatic Δt control: maximum timestep.

param 46 *default value = 1.00*

Automatic Δt control: maximum fractional change in temperature per timestep.

param 49 *default value = 5.00*

Automatic Δt control: maximum value of temperature equation eigenvalue times Δt.

param 55 *default value = 1.00e-2*

Convergence control on temperatures: timesteps will be iterated until the maximum fractional change in electron and ion temperatures between two successive iterations falls below param(55).

param 56 *default value = 1.00e-2*

Convergence control on populations: timesteps will be iterated until the maximum fractional change in those iso-sequence populations with sufficient population between two successive iterations falls below param(56) - see param(60).

param 57 *default value = 1.00e-2*

Convergence control on line strengths: timesteps will be iterated until the maximum fractional change in line strengths between two successive iterations falls below param(57).

param 60 *default value = 1.00e-10*

Minimum fractional population in iso-sequence necessary for considering convergence. Iso-sequences with smaller fractional population than param(60) will not affect timestep iterations.

Output:

Edits defined in the generator will appear in the *.tbl*, *.plt*, and *.ult* files, while information requested with a **dump** command will appear in a *.dxx* file. Edits and dumps will be produced upon initialization (cycle 0), termination and at intervals controlled by switch(30) and param(40). (The exception to this is that a dump will not be produced upon initialization from a restart dump). Information for time-dependent and time-integrated edits will be saved for every timestep and sent to the appropriate files upon termination.

Restart dumps and spectral dumps can be controlled independently from the edits and dumps, via switch(65) and param(87) for restart dumps, and switch(73) and param(92) for spectral dumps. These dumps can also be produced interactively.

Spectral calculations are required to produce information for edits and (possibly) dumps, but not for restart dumps or spectral dumps. The minimum frequency of spectral calculations is also controlled by switch(30) and param(40). However, this may not update information frequently enough for time-dependent and time-integrated edits. More frequent spectral calculations can be requested via switch(72) and param(91) for this purpose.

Relevant switches and parameters:

switch 10 *default value = 0*

Echo output to *.tbl* file to screen during run if switch(10)≠0.

switch 11 *default value = 0*

Choose file(s) for writing out plots
No files if switch(11)<0.
Default file (PDB *.ult* file if possible, ascii *.plt* file otherwise) if switch(11)=0
Both files (*.ult* and *.plt*) if switch(11)=1
Ascii *.plt* file only if switch(11)=2

switch 12 *default value = 0*

Rename *.ult* and *.plt* file(s) to avoid overwriting existing files if switch(12) ≠0. New names will have familied suffixes, e.g. *.u00, .u01, .u02*, etc.

switch 26 *default value = 0*

Tables in *.tbl* file will have 3 significant figures if switch(26)=0, 6 significant figures if switch(26)≠0.

switch 30 *default value = 1000*

Number of timesteps between edits / dumps.

switch 61 *default value = 0*

Include list of photoexcitations, sorted by energy, in *.tbl* file for each atomic model if switch(61)≠0.

switch 65 *default value = 1000*

Number of timesteps between restart dumps.

switch 72 *default value = 1000*

Number of timesteps between spectral calculations.

switch 73 *default value = 1000*

Number of timesteps between spectral dumps.

switch 75 *default value = 0*

Do not make initial timestep edits if switch(75)>0, do not make any time-independent edits if switch(75)<0.

switch 99 *default value = 22*

Window tiling for XGRAFIX. The 10's digit denotes the number of windows horizontally, while the 1's digit denotes the number of windows vertically. Windows beyond those which fill the tiling will overlap the tiled windows.

param 40 *default value = 1.00e+10 (sec)*

Time interval between edits / dumps.

param 82 *default value = 0.00 (eV)*

Instrument broadening (fwhm) to be used in each broadened spectral edit for which a value was not specified.

param 83 *default value = 0.00*

Cosine of viewing angle with respect to z-axis for 1-d cylindrical *jsparea* edits.

param 84 *default value = 1.00 (cm)*

Length of cylinder to be used for calculating 1-d cylindrical *jsparea* edits.

param 87 *default value = 1.00e+10 (sec)*

Time interval between restart dumps.

param 91 *default value = 1.00e+10 (sec)*

Time interval between spectral calculations.

param 92 *default value = 1.00e+10 (sec)*

Time interval between spectral dumps.

Neutral Diffusion:

The neutral diffusion package couples the atomic kinetics equations with a set of diffusion equations for neutral levels, allowing an approximate treatment of neutral recycling in radiative divertors. The set of equations for the level populations *y* are

$$\frac{\mathrm{d}y}{\mathrm{dt}} + \nabla \cdot \frac{1}{m_{ion}\upsilon_x} \nabla (kTy) = \mathbf{A}y$$

where m_{ion} is the ion mass, $\nu_x = 2 \times 10^{-8}\, n^4\, n_{\mathrm{ion}}$ is the charge-exchange frequency, n is the principal quantum number of the neutral level, and n_{ion} is the (non-neutral) ion density, T is the ion temperature, and **A** is a matrix of rate matrices (for each node). Non-neutral levels do not experience diffusion, but are coupled to the neutral levels through the rate equations.

Boundary conditions for the diffusion equations are assumed to be zero flux (i.e. reflecting) for all but the ground state level of the neutral ions. Boundary conditions for the ground state level may be specified with a series of **boundary** commands with the *package* argument equal to "divertor", i.e.

boundary divertor *type ir value* or
boundary divertor *type k1 k2 l1 l2 value*

Specifying a *type* of "recycling" produces a boundary flux = -*value* n_{ion} c_s, where $c_s = [k(T_e+T_{ion})/m_{ion}]^{½}$. Other possibilities for *type* include "value", "gradient", "flux", "reflecting" and "mirror".

Relevant switches and parameters:
(note that the neutral diffusion package has its own set of switches and parameters)

switch 60 *default value = 0*

Couple kinetics and neutral diffusion if switch(60)≠0. Note that dswitch(1) must also be set ≠ 0.

dswitch 1 *default value = 0*

Main control: neutral diffusion calculations will be done only if dswitch(1) ≠ 0. Note that switch(60) must also be set ≠ 0.

dswitch 2 *default value = 0*

Populations will be calculated assuming steady-state diffusion if dswitch(2) = 0, or assuming time-dependent diffusion if dswitch(2) ≠ 0.

NOTE: The iterative solution works currently works much better if both the diffusion and atomic kinetics are time-dependent.

dswitch 5 *default value = 4*

Control choice of preconditioning for gmres solver:
no preconditioning if dswitch(5) = 0
solve diffusion equations if dswitch(5) = 1
solve rate equations if dswitch(5) = 2
solve rate equations, then diffusion equations if dswitch(5) = 3
solve diffusion equations, then rate equations if dswitch(5) = 4

dswitch(5) = 4 is the preferred setting. However, this may not work for all problems. dswitch(5) = 2 is the most robust setting.

dswitch 6 *default value = 0*

Control choice of solver for coupled diffusion/kinetics equations:

1-d: use tridiagonal direct solver if dswitch(6) = 0.
use gmres if dswitch(6) ≠ 0.
2-d: use gmres for any value of dswitch(6).

dswitch 7 *default value = 100*

The maximum number of orthogonalizations used by gmres. The total number of orthogonalizations (including restarts) necessary for convergence is problem dependent. Using kinetics preconditioning (dswitch(5)=2), gmres should converge to near machine precision is a number of orthogonalizations equal to the number of spatial zones.

dswitch 8 *default value = -1*

The maximum number of restarts used by gmres. Set this to -1 for no restarts as 0 will result in the (default) value of 10 restarts.

dswitch 9 *default value = 0*

Set constraint for solving coupled diffusion/kinetics equations:
conserve mass density at each node if dswitch(9) = 0
conserve ion density at each node if dswitch(9) ≠ 0.

dswitch 10 *default value = 100*

The maximum number of iterations used by iccg (used to solve 2-d diffusion equations during preconditioning for gmres).

dswitch 11 *default value = 0*

Diffuse product of ion density times temperature if dswitch(11) = 0.
Diffuse ion density if dswitch(11) ≠ 0 (i.e. move kT out of the gradient operator).

dswitch 12 *default value = 0*

Control printing of gmres diagnostics:
no printing if dswitch(12) = 0
print to *.tbl* file if dswitch(12) > 0
print to screen file if dswitch(12) < 0.

dswitch 13 *default value = 0*

Control printing of iccg diagnostics:
no printing if dswitch(12) = 0
print to *.tbl* file if dswitch(12) > 0
print to screen file if dswitch(12) < 0.

dparam 2 *default value = 1.0*

Multiplier for diffusion coefficients.

dparam 3 *default value = 0.0*

If dparam(3) ≠ 0.0, use dparam(3) as the total ion density for each node in the problem.

dparam 4 *default value = 1.00e-2*

Convergence criterion for gmres iterations.

dparam 5 *default value = 1.00e-2*

Convergence criterion for iccg iterations.

Summary of Switches

The following switches should not be used:

1 - 7, 9, 13 - 16, 18, 19, 21 - 24, 27, 43, 45 - 48, 58, 59, 64, 69 - 71, 76, 95, 96, 100, 103

Switch	Default	
8	1	choose angle set for radiation transfer 1-d: use gaussian angles if =0, double-gaussian angles if ≠0 2-d: use Carlson A angles if =0, Carlson B angles if =1 product ray set based on gaussian if =2, double-gaussian if >=3 3-d: same as 2-d
10	0	write *.tbl* file to screen during run
11	0	choose files to use for writing plots <0: no files 0: default file (PDB *.ult* file if possible, ascii *.plt* file otherwise) 1: both files (*.ult* and *.plt*) 2: ascii *.plt* file only
12	0	rename *.ult* and *.plt* files to avoid overwriting existing files if ≠0
17	0	include line opacities and emissivities in continuum quantities for 0: no transitions >0: all transitions not identified as lines <0: all transitions
20	1	Radiation control for kinetics 0: LTE 1: NLTE 2: NLTE w/ black-body photons at T=Te 3: NLTE w/ black-body photons at T=Tr
25	1	Steady-state kinetics if =0 time-dependent kinetics if >0 no kinetics if <0
26	0	3-place tables if =0 6-place tables if ≠0
28	0	Initialization control <0: LTE at fixed electron density 0: LTE at fixed ion density 1: steady-state 2: none
29	2	timestep control 0: use xfile timesteps 1: use constant timesteps (param(41)) 2: use dynamic timesteps
30	1000	number of timesteps between snapshots
31	0	temperature calculation 0: none >0: time-dependent <0: steady-state
32	1	include line intensities in photoionizations if >0, in photoexcitations if >1
33	0	use escape factors for all photoexcitations if >0, for lines only if <0

		±1: static ±2: Sobolev ±3: interpolation ±4: generalized
34	0	symmetry: 1-D 0: non-symmetric <>0: symmetric about r=0.
		2-D symmetry: 2-digit integer kl 3-D symmetry: 3-digit integer klm for each digit *x* (k,l,m) *x* = 0: non-symmetric in the *x*-direction *x* = 1 symmetric about *x* = 1 *x* = 2: symmetric about *x* = *x*max *x* = 3: symmetric about *x*=1 and *x*=*x*max
35	0	turn off bremsstrahlung if ≠0
36	0	do continuum transfer if ≠0 1-d: use Feautrier formalism if =1, integral formalism otherwise
37	0	do line transfer if ≠0
38	0	assume constant continuum quantities and use symmetric line profiles when possible if ≠0
39	3	max. iterations for scattering in continuum transfer
40	13	max. iterations for approx. operator line transfer
41	4	iterations before acceleration in approx. operator line transfer
42	0	produce node-centered mesh if zero, zone-centered mesh if non-zero
44	0	max. iterations for single timestep
49	0	include electron thermal conduction if ≠0 if 2-d: ±1: iccg ±2: ilur ±3: gmres w/ diagonal preconditioning ±4: gmres w/ iccg preconditioning ±5: gmres w/ ilur preconditioning other: gmres w/o preconditioning <0: print convergence diagnostics to screen
50	0	include flux limiting on conduction if ≠0
51	0	set coulomb logarithm, conduction z-term to 1. if ≠0
52	0	include Stark broadening <0: for all photoexcitations >0: for all lines
53	0	do detailed spectral calculation <-1: no spectral calculation -1: calculate opacities and emissivities only 0: specified edit rays only 1: all rays >1: all rays including electron scattering
54	0	<0: include only special transitions (lines, Stark transitions) in spectrum 0: include continuum in spectrum, interpolate from continuum groups 1: do not include continuum in spectrum 2: include continuum in spectrum, recalculate on spectral groups >2: convolve continuum with selected line shapes last distinct line (Inglis-Teller limit) using Stark width if =3 last distinct line in model using calculated width if =4 transition from **kernel** command if >4
55	0	do continuum lowering if ≠0

56	2	number of LTE iterations w/continuum lowering
57	0	calculate natural width of photoexcitations including >=0: both level lifetimes <0: upper level lifetime only ———— 0, ±1: all transitions (except ground states) ±2: no recombinations (except ground states) ±3: excitations only (except ground states) ±4: include ground state lifetimes
58	0	Update material properties (ne, zbar, etc.) after kinetics calculation <0: with fixed electron density 0: with fixed ion densities >0: not at all
60	0	couple kinetics to neutral diffusion if ≠0
61	0	dump sorted photoexcitation list to .tbl file if ≠0
62	0	include electron scattering as loss in line transfer if ≠0
63	0	include doppler shifts in spectral calculation if ≠0
65	1000	number of timesteps between restart dumps
66	100	maximum number of iterations of iccg
67	10	maximum number of orthogonalizations for gmres
68	4	maximum number of restarts for gmres (set to -1 for no restarts)
72	1000	number of timesteps between spectral calculations
73	1000	number of timesteps between spectral dumps
74	0	skip 2-d transfer calculations for timesteps with rayset errors if ≠0
75	0	do not do an initial timestep edit if >0, do not do any time-independent edits if <0
77	0	do not allow negative continuum opacities if ≠0
78	0	assume Voigt profile for escape factors if =0, Doppler profile if ≠0
79	0	include continuum opacities in escape factors if = 0
80	0	specify 1-d geometry used for escape factors 0: problem geometry 1: planar 2: cylindrical 3: spherical
81	0	use double-sided escape factors if = 0, single-sided if ≠0
82	0	use integrated column densities for escape factors if = 0, use local densities if ≠0
83	0	include additional energy sources in temperature calculation if ≠0
84	0	zone-center conduction coefficients if ≠0
85	0	use special scaling of initial densities and temperatures if ≠0
86	50	number of microfield values used in TOTAL
87	0	group transitions for TOTAL 0: individually 1: by principal quantum number 2: by isosequence
88	0	calculate real eigenvalues in TOTAL if =0, complex eigenvalues if ≠0
89	0	include line polarization shifts if ≠0
90	0	use xfile for initialization only if ≠0
91	0	use updated populations for opacities, emissivities if ≠0
92	1	calculate spectral opacities <0: read from dump files 0: in chunks >0: all together
93	0	include re-entrant radiation (in 2-d and 3-d) if ≠0
94	0	use scratch file for 2-d and 3-d ray information if ≠0
97	0	zone number for individual kinetics calculation

98	0	recalculate line profiles for each angle if ≠0
99	22	window tiling for XGRAFIX (10's - horizontal, 1's - vertical)
101	0	include Debye shielding for collisional rates if ≠0
102	0	fix level populations from xfile if ≠0
104	0	overlap radiation transfer work spaces if ≠0
105	10	maximum size of dump files (MB)
106	10	maximum size of spectral dump files (MB)
107	10	maximum size of sensitivity dump files (MB)

Summary of Parameters

The following parameters should not be used:

1 - 4, 6 - 10, 12 - 27, 29 -39, 47, 58 - 60, 63

Param	Default	
5	1.00	multiplier for electron scattering
11	0.00	free-free Gaunt factor if non-zero
28	1.00e-02	convergence value for continuum radiation intensities
40	1.00e+10	time interval between snapshots (sec)
41	1.00e-12	initial timestep (sec)
42	0.20	max. change in zbar per timestep
43	1.50	max. fractional increase in timestep
44	1.00e-15	minimum timestep (sec)
45	1.00	maximum timestep (sec)
46	1.00	max. fractional increase in temperature per timestep
48	1.00	multiplier for electron-ion coupling
49	5.00	temperature timestep control
50	10.0	min. # of doppler widths in CRD line wings
51	1.00e-03	convergence value for approx. operator line transfer
52	4.00	max. tau ratio for 2nd-order fit
53	0.00	column density for escape factors (g/cm^2 or $1/cm^2$)
54	1.00e-02	min. optical depth for escape factors
55	1.00e-02	convergence value for temperatures
56	1.00e-02	convergence value for iso-sequence populations
57	1.00e-02	convergence value for line strengths
58	0.00	opening angle for wedge geometry (radians)
59	0.00	r-offset for 1-d x-file positions
60	1.00e-10	threshold fractional population for consideration in convergence test
61	0.00	threshold fractional population for inclusion of iso-sequence (retain initial window if =0.)
62	1.00	multiplier on limit to line-width contribution of coll. excitations
64	1.00	multiplier on electron thermal conduction
65	1.00	multiplier on flux limiter
66	1.00	multiplier on coes rates
67	1.00	multiplier on coll excite rates
68	1.00	multiplier on coll ionize rates
69	1.00	multiplier on cois rates
70	1.00	multiplier on phxs cross-sections
71	1.00	multiplier on phis cross-sections
72	1.00	multiplier on augxs cross-sections
73	1.00	multiplier on total augis rates
74	1.00	multiplier on augis rec rates
75	1.00	multiplier on augis rad rates
76	1.00	multiplier on augis aug rates
77	1.00	multiplier on augis cex rates
78	1.00e4	max. # of Lorentz widths included in spectra
79	2.00	exponent in degeneracy lowering formula

80	1.00e-02	min. optical depth for linearizing line transfer
81	0.50	limit on line widths as a fraction of line energy (for spectra)
82	0.00	instrument broadening (fwhm) for spectral edits (eV)
83	0.00	cosine of viewing angle w.r.t. z-axis for cylindrical *jsparea* edits
84	1.00	length for cylindrical *jsparea* edits (cm)
85	1.00	multiplier for opacity in spectral calculation
86	1.00	multiplier for emissivity in spectral calculation
87	1.00e+10	time interval between restart dumps (sec)
88	1.00e-4	convergence value for iccg or gmres calculations (for conduction)
89	1.00	multiplier for continuum lowering
90	0.00	minimum doppler width for spectral calculations (eV)
91	1.00e+10	time interval between spectral calculations (sec)
92	1.00e+10	time interval between spectral dumps (sec)
93	1.00	multiplier for line contributions to spectral opacities, emissivities
94	0.00	coefficient for energy source term
95	0.00	minimum electron density for transition rates (fraction of ion density)
96	0.50	relaxation parameters for electron density iterations

Atomic Physics Input

Atomic data must be included in an ascii datafile for each atomic model specified in the generator. The data defines processes in one direction only - from a lower state to an upper state. The inverse processes will be calculated to be consistent with detailed balance. Transitions which do not correspond to defined levels are ignored (with a warning). Blank lines can be freely interspersed within the datafile. Lines beginning with *c* are interpreted as comments. Arguments within brackets are optional. Additional fields on any command line are permitted.

File organization:

idatname *name*

This command is ignored by CRETIN.

data *type*

The **data** command specifies the beginning of a section of levels or fitted atomic data. The type may be model, *coes, coll excite, coll ionize, phxs, phis, augxs, augis* or *cois*. There may be only one data section for each type. The type *phxs1* is also acceptable, but there cannot be sections for both *phxs* and *phxs1*.

end

This command signifies the end of a section of levels or fitted data.

Model specification:
model

enot *iso nameiso energy*

This command defines an isoelectronic sequence *iso* with name *nameiso* and ionization potential *energy* (eV). The isoelectronic sequences include 0 for a bare nucleus, 1 for a hydrogenic ion, 2 for a helium-like ion, etc.

elev *iso i namelev g energy [... n]*

This command defines a level within an isoelectronic sequence, where *iso* identifies the isoelectronic sequence, *i* is the level number within the sequence, *namelev* is an identifier for the level (which is usually ignored by CRETIN but must be present), *g* is the statistical weight, and *energy* is the energy of the level relative to the ground state of the sequence. The principal quantum number *n* must be the last field on the line.

If the identifier *namelev* includes the sequence "*n*X", where *n* is the principal quantum number, X will be interpreted to yield the angular momentum quantum number *l*, e.g. X="s" or "S" for *l*=1, X="p" or "P" for *l*=2. The first occurrence of *n* in *namelev* must be in the sequence "*n*X".

c shell *k-index l-index m-index …*

This command tells CRETIN to interpret various fields on the **elev** command as shell occupations. The *k-index* field contains the occupation of the k-shell, the *l-index* field contains the occupation of the l-shell, etc.

This interpretation remains in effect for all succeeding **elev** commands until another **shell** command is encountered. The shell occupations are currently used only for edit purposes. Note that the **shell** command must be preceded by a **c**, so that other codes will interpret this as a comment.

Electron collisional excitation/ionization:

coes, coll excite, coll ionize

d *iso1 i1 iso2 i2 a0 a1 a2 a3 n de tevmin tevmax*

This command defines an electron collisional excitation or ionization process from lower level (*iso1*,*i1*) to upper level (*iso2*,*i2*). *a0*, *a1*, *a2*, *a3*, *n*, and *de* are fitting parameters as defined below. *tevmin* and *tevmax* are the minimum and maximum electron temperatures (eV) over which the rate coefficients were fitted.

The rate coefficient sigma (cm^3/sec) for a transition of energy Δe and an electron temperature *tev* is defined by

$$\text{sigma} = 1.58 \times 10^{-5}\, n\, (p(b)/b)\, e^{-b} / tev^{1.5}$$

where

$$b = \begin{cases} \Delta e / tev & tevmin < tev < tevmax \\ de / tevmin & tevmin > tev \\ de / tevmax & tev > tevmax \end{cases}$$

and

$$\ln p(b) = a0 + a1\, (\ln b) + a2\, (\ln b)^2 + a3\, (\ln b)^3.$$

If requested, the rate coefficient is then modified for Debye shielding in the following manner:

$$1/\text{sigma (shielded)} = 1/\text{sigma} + 1/(v_e D^2)$$

where v_e is the electron thermal velocity and D is the Debye length.

Photoexcitation:

phxs or *phxs1*

d *iso1 i1 iso2 i2 f [lambda]*

This command defines a photoexcitation from lower level (*iso1*,*i1*) to upper level (*iso2*,*i2*). *f* is the absorption oscillator strength (dimensionless). If *lambda* is present, it is taken to be the wavelength of the transition (in Angstroms), otherwise the energy of the transition is calculated from the energy levels of the model.

Photoionization:

phis

d *iso1 i1 iso2 i2 a0 a1 a2 a3 n de emin emax*

This command defines a photoionization process from lower level (*iso1*,*i1*) to upper level (*iso2*,*i2*). *a0*, *a1*, *a2*, *a3*, *n*, and *de* are fitting parameters as defined below. *emin* and *emax* are the minimum and maximum photon energies (eV) over which the fit is valid.

The cross section sigma (cm^2) for a photon energy *e* (eV) is defined by

sigma = 10^{-18} *n* p(b) (13.606/*de*)

where

b = *e* / *de*

and

ln p(b) = *a0* + *a1* (ln b) + *a2* (ln b)^2 + *a3* (ln b)^3.

Auger Excitation:

augxs

d *iso1 i1 iso2 i2 f lambda [a] [g]*

This command defines a photoexcitation from lower level (*iso1,i1*) to a doubly excited state with a transition energy corresponding to the wavelength *lambda* (Angstroms). The doubly excited state decays through an Auger process to level (*iso2,i2*). *f* is the absorption oscillator strength (dimensionless). If *a* is present, it is the Auger decay rate, otherwise the decay rate is assumed to be 10^{18}/sec. If *a* is negative, an infinite decay rate is assumed. If *g* is present, it is the statistical weight of the doubly excited state, otherwise the statistical weight is 1.

d *iso1 i1 iso2 i2 a*

This command defines an Auger decay from upper level (*iso2,i2*) to lower level (*iso1,i1*) with decay rate *a* (1/sec).

Auger Ionization:

augis

d *iso elev g*

This command begins the specification of a set of Auger processes by defining a doubly excited state in isoelectronic sequence *iso* with an energy *elev* above the ground state of that sequence and statistical weight *g*. The doubly excited state can participate in a number of processes connecting levels in isoelectronic sequences *iso*-1, *iso*, and *iso*+1 as defined by any number of the following commands.

d *iso1 i1 iso2 i2 a0 a1 a2 a3 n de emin emax [g]*

This is an alternative command which also begins the specification of a set of Auger processes where (*iso1,i1*) is the starting level which has an inner shell electron that can be photoionized to yield a doubly excited state an energy *emin* above the starting level and statistical weight *g* (or 1 if *g* is absent). The photoionization cross section is defined as in the *phis* section. The previous form of the **d** command is preferred for new models.

rec *iso1 i1 a0 a1 a2 a3 n de emin emax*

This command defines a radiative ionization from (*iso1,i1*) to the doubly excited state with a cross section defined as in the *phis* section. This cannot be used with the second form of the **d** command.

rad *iso2 i2 de f*

This command defines a radiative decay from the doubly excited state to (*iso2*,*i2*) with transition energy de and absorption oscillator strength *f*.

aug *iso2 i2 rate*

This command defines an Auger decay from the doubly excited state to (*iso2*,*i2*) with decay rate *rate* (1/sec).

cex *iso1 i1 a0 a1 a2 a3 n de tevmin tevmax*

This command defines a collisional excitation or ionization from (*iso1*,*i1*) to the doubly excited state with a rate defined as in the *coes* section.

done

This command finishes the specification of the set of processes begun by a **d** command.

Ion collisional excitation/ionization:

cois

per *z iso*

This command specifies that all processes until the next **per** command refer to interactions with ions of atomic number *z* and isoelectronic sequence *iso*.

d *iso1 i1 iso2 i2 a0 a1 a2 a3 n de tivmin tivmax*

This command defines an ion collisional excitation or ionization process from lower level (*iso1*,*i1*) to upper level (*iso2*,*i2*). *a0*, *a1*, *a2*, *a3*, *n*, and *de* are fitting parameters as defined below. *tivmin* and *tivmax* are the minimum and maximum ion temperatures (eV) over which the rate coefficients were fitted.

The rate coefficient sigma (cm^3/sec) for a transition of energy Δe and an ion temperature *tiv* is defined by

$$\text{sigma} = 1.58 \times 10^{-5}\, n\, (p(b)/b)\, e^{-b} / tiv^{1.5}$$

where

$$b = \begin{cases} \Delta e / tiv & tivmin < tiv < tivmax \\ de / tivmin & tivmin > tiv \\ de / tivmax & tiv > tivmax \end{cases}$$

and

$$\ln p(b) = a0 + a1\, (\ln b) + a2\, (\ln b)^2 + a3\, (\ln b)^3.$$

Xfile Input

Various quantities can be specified in supplementary data files (xfiles), as defined below. There may be any number of xfiles, but xfiles beyond the first one can contain only *pbins* or *pbcgs* sections and are assumed to have the same group structure as the first xfile. Blank lines can be freely interspersed within the xfile. Lines beginning with *c* are interpreted as comments. If only one problem time is present in the file, the data will be used for initialization at generation time only. Commands ending in "**3d**" are identical to those ending in "**2d**" – either form may be used for either dimensionality.

problem *name time itime ir1 ir2*

Specify the time *time* (sec) for which the following data applies and the number of nodes ($n = ir2\text{-}ir1+1$) for which data will be supplied. The remaining fields are necessary but are ignored by CRETIN.

e *ng*

This command is followed by *ng* energies (eV) to define the group structure for continuum radiation. The energies appearing here are the group upper boundaries. The lower boundary of the first group is zero. This command should appear only in the first section of the first xfile.

pbins *ir*

This command is followed by *ng* photon energy densities (jerks/keV/cm^3) for node *ir*. If the energy densities are not specified for each node in the problem, the information from the first node will be broadcast to all other nodes or used as boundary information where appropriate. This command cannot coexist with the **pbcgs** command.

pbcgs *ir*

This command is followed by *ng* photon energy densities (erg/Hz/cm^3) for node *ir*. If the energy densities are not specified for each node in the problem, the information from the first node will be broadcast to all other nodes or used as boundary information where appropriate. This command cannot coexist with the **pbins** command.

r

This command is followed by *n* 1-d positions (cm).

x2d or **x3d**

This command is followed by *n* 2-d (x- or r-) or 3-d (x) positions (cm).

y2d or **y3d**

This command is followed by *n* 2-d (y- or z-) or 3-d (y) positions (cm).

z2d or **z3d**

This command is followed by *n* 2-d (y- or z-) or 3-d (z) positions (cm).

u

This command is followed by *n* 1-d velocities (cm/sec).

u2d or **u3d**

This command is followed by *n* 2-d (x- or r-) or 3-d (x) velocities (cm/sec).

v2d or **v3d**

This command is followed by *n* 2-d (y- or z-) or 3-d (y) velocities (cm/sec).

w2d or **w3d**

This command is followed by *n* 2-d (z- or f-) or 3-d (z) velocities (cm/sec).

durdr

This command is followed by *n* 1-d velocity gradients (1/sec).

d

This command is followed by *n* densities (g/cm^3).

te

This command is followed by *n* electron temperatures (eV).

ti

This command is followed by *n* ion temperatures (eV).

tr

This command is followed by *n* radiation temperatures (eV).

bx

This command is followed by *n* magnetic field x- (1-d planar or 2-d xy or 3-d) or r- (1-d cylindrical, spherical or 2-d rz) components (G).

by

This command is followed by *n* magnetic field field y- (1-d planar or 2-d xy or 3-d) or theta (1-d spherical) or z- (1-d cylindrical or 2-d rz) components (G).

bz

This command is followed by *n* magnetic field field z-(1-d planar or 2-d xy or 3-d) or phi- (1-d cylindrical,spherical or 2-d rz) components (G).

bfield

This command is followed by *n* magnetic field magnitudes (G).

regmap *kmax [lmax] [mmax]*

This command is followed by *n* integers identifying each node with a region. Currently, CRETIN uses this information only to identify the values of *kmax* (and *lmax*, *mmax*, for a 2- or 3-d mesh). These values must be consistent with *n* = *kmax* or *n* = *kmax* x *lmax* or *n* = *kmax* x *lmax* x *mmax*.

logmesh *kmax lmax*

This command is equivalent to the **regmap** command.

done

This command terminates the data associated with a given time or **problem** command.

The following 6 commands allow the user to define time-dependent element and electron number densities for each node in the problem. The **znuc** and (optionally) **anuc**, **isonuc** and **i** commands need only be present once in the xfile to identify the elements for which number densities will be specified. If the atomic number (and atomic weight, if present) match the specification in an **atoms** command in the generator and are specified in a **source** command, the number densities from the xfile will be used for that element or isosequence. Elements are not restricted to the regions where they were specified in the generator. If an element appears at a node where it was previously absent, the populations are initialized to steady-state conditions appropriate to that node. If the **znuc** command is present, either the **ne**, **ni** and **zbar** commands must also be present, or the **isonuc** and **ni** commands must also be present.

znuc *nnuc*

This command is followed by *nnuc* atomic numbers.

anuc *nnuc*

This command is followed by *nnuc* atomic weights.

isonuc *nnuc*

This command is followed by *nnuc* isosequence specifications.

inuc *nnuc*

This command is followed by *nnuc* level specifications.

ne

This command is followed by *n* electron densities ($1/cm^3$).

ni *inuc*

This command is followed by *n* number densities for the element (or isosequence) *inuc* defined with the **znuc** (or **znuc** and **isonuc**) [and **anuc**] command[s] ($1/cm^3$).

zbar

This command is followed by *n* average ion charges.

The following 2 commands allow the user to define time-dependent electron number distributions whose temperatures can be different from that of the main electron distribution.

tehot *ihot*

This command is followed by *n* temperatures for electron distribution *ihot* (eV).

nehot *ihot*

This command is followed by *n* number densities for electron distribution *ihot* (eV).

Edits

All edits in CRETIN are specified by an edit name and four indices e.g.,
edit i1 i2 i3 i4 .
The first index, *i1*, specifies either the model (*iz*), line (*iline*), transition (*itr*), spectral-group (*is*), or xfile (*ix*). The second index, *i2*, specifies the spatial node (*ir*), k-line (*k*) or l-line (*l*). The third index, *i3*, specifies either the continuum group (*igp*), spectral energy group (*isp*), line energy bin (*ifr*) or isoelectronic sequence (*iso*). The fourth index, *i4*, specifies either an atomic level (*i*) or directional ray (*iray*). Trailing indices which are zero may be omitted.

If an edit runs over a particular index (e.g., editing a quantity for all nodes), that index should be zero. The exception is if an edit runs over a k-line or (l-line), that index should specify the k-line (or l-line). There are also three special cases for indices: *ir* = -1 specifies the last node, *ifr* = -1 specifies the bin at line center and *isp* = -*iline* specifies the spectral energy group corresponding to the center of line *iline*.

Units for the edits are mostly c.g.s. with a few exceptions. Some energies and temperatures are in eV, line intensities are in photons/mode (as well as c.g.s.). Continuum and spectral radiation quantities are in c.g.s., but these units can be modified separately for each edit with a **rad-units** command.

CRETIN does NOT check edits requests for correctness or consistency. Undefined edits will return zero values. Non-zero values for unused indices will not affect the results -- this does not include the (usually-) zero value for the running index.

Independent variables for *xvar*:

1. cycle, iter, time
2. ir, r, cdens, x2d, y2d, z2d, x3d, y3d, z3d, xy, k, kr, kx, ky, kz, l, lr, lx, ly, lz, m, mr, mx, my, mz
3. ifr, energy, freq, wvl, ebins, fbins, wbins, ifrline, evline, isp, sp_energy, sp_freq, sp_wvl
4. iso, level, elev

Time:

cycle

Timestep number.

iter

Timestep number.

itime

Timestep number.

time

Time (sec).

dtime

Timestep (sec).

dtfrac

Fractional timestep.

Mesh properties:

ir *0 ir*

Node index.

nnode

Total number of nodes in mesh.

nzone

Total number of zones in mesh.

kmax

Number of nodes in k-direction.

lmax

Number of nodes in l-direction.

mmax

Number of nodes in m-direction.

r *0 ir*
xy *0 m*

Position of node *ir* (cm). NOTE: For a 2-d problem, "**xvar** r" is expanded to "**xvar** x2d" and "**xvar** y2d". For a 3-d problem, "**xvar** r" is expanded to "**xvar** x3d", "**xvar** y3d" and "**xvar** z3d" and "**xvar** xy" is expanded to "**xvar** x3d" and "**xvar** y3d".

x2d *0 ir*
x3d *0 ir*

Position of node *ir* in x- (or r-) direction (cm).

y2d *0 ir*
y3d *0 ir*

Position of node *ir* in y- (or z-) direction (cm).

z2d *0 ir*
z3d *0 ir*

Position of node *ir* in z-direction (cm).

cdens *0 ir*

Column density from node 1 to node *ir* (g/cm^2).

vel *0 ir*

Speed at node *ir* (cm/sec).

dvdr *0 ir*

Magnitude of velocity gradient at node *ir* (1/sec).

u2d *0 ir*
u3d *0 ir*

Velocity component in x- (or r-) direction for node *ir* (cm/sec).

v2d *0 ir*
v3d *0 ir*

Velocity component in y- (or z-) direction for node *ir* (cm/sec).

w2d *0 ir*
w3d *0 ir*

Velocity component in z-direction for node *ir* (cm/sec).

For the following quantities, the second index should specify the k-line *k* or the l-line *l* if the quantity is an independent x-variable, and should specify the node *ir* if the quantity is a dependent y-variable. The third index should specify the m-line *m* if the quantity is an independent x-variable for a 3-d problem. If the quantity is a dependent y-variable, the k-, l-, and m-indices will be obtained from the x-variables if necessary.

k *0 (l,ir) (m)*

k-line index along l-line *l* (and m-line *m*, if 3-d) or for node *ir*.

kr *0 (l,ir) (m)*

Distance along l-line *l* (and m-line *m*, if 3-d) from node at *(1,l,m)* to node *ir* at *(k,l,m)* (cm).

kx *0 (l,ir) (m)*

Distance along x- (or r-) axis (for l-line *l*, and m-line *m*, if 3-d) from node at *(1,l,m)* to node *ir* at *(k,l,m)* (cm).

ky *0 (l,ir) (m)*

Distance along y- (or z-) axis (for l-line *l*, and m-line *m*, if 3-d) from node at *(1,l,m)* to node *ir* at *(k,l,m)* (cm).

kz *0 (l,ir) (m)*

Distance along z-axis (for l-line *l*, and m-line *m*, if 3-d) from node at *(1,l,m)* to node *ir* at *(k,l,m)* (cm).

l *0 (k,ir) (m)*

l-line index for k-line *k* (and m-line *m*, if 3-d) or for node *ir*.

lr *0 (k,ir) (m)*

Distance along k-line *k* from node at *(k,1,m)* to node *ir* at *(k,l,m)* (cm).

lx *0 (k,ir) (m)*

Distance along x- (or r-) axis (for k-line *k*, and m-line *m*, if 3-d) from node at *(k,1,m)* to node *ir* at *(k,l,m)* (cm).

ly *0 (k,ir) (m)*

Distance along y- (or z-) axis (for k-line *k*, and m-line *m*, if 3-d) from node at *(k,1,m)* to node *ir* at *(k,l,m)* (cm).

lz *0 (k,ir) (m)*

Distance along z-axis (for k-line *k*, and m-line *m*, if 3-d) from node at *(k,1,m)* to node *ir* at *(k,l,m)* (cm).

m *0 ir*

m-line index along k-line *k* and l-line *l* (specified by *ir = kmax*l+k*) or for node *ir*.

mr *0 ir*

Distance along m-line *m* from node at *(k,l,1)* to node *ir* at *(k,l,m)* (cm).

mx *0 ir*

Distance along x- (or r-) axis (for k-line *k* and l-line *l*) from node at *(k,l,1)* to node *ir* at *(k,l,m)* (cm).

my *0 ir*

Distance along y- (or z-) axis (for k-line *k* and l-line *l*) from node at *(k,l,1)* to node *ir* at *(k,l,m)* (cm).

mz *0 ir*

Distance along z-axis (for k-line *k* and l-line *l*) from node at *(k,l,1)* to node *ir* at *(k,l,m)* (cm).

Plasma properties:

ne *0 ir*

Electron density at node *ir* ($1/cm^3$).

dnedt *0 ir*

Derivative w.r.t. time of electron density at node *ir* ($1/cm^3$).

nion *0 ir*

Ion density at node *ir* (1/cm^3).

ntot *0 ir*

Particle density at node *ir* (1/cm^3).

rho *0 ir*

Mass density at node *ir* (g/cm^3).

te *0 ir*

Electron temperature at node *ir* (K).

tev *0 ir*

Electron temperature at node *ir* (eV).

ti *0 ir*

Ion temperature at node *ir* (K).

tiv *0 ir*

Ion temperature at node *ir* (eV).

tr *0 ir*

Radiation temperature at node *ir* (K).

trv *0 ir*

Radiation temperature at node *ir* (eV).

tehot *i ir*

Temperature of ith electron distribution at node *ir* (eV).

nehot *i ir*

Number density of ith electron distribution at node *ir* (1/cm^3).

emat *0 ir*

Material energy density at node *ir* (erg/cm^3).

emat_e *0 ir*

Electron energy density at node *ir* (erg/cm^3).

emat_i *0 ir*

Ion energy density at node *ir* (erg/cm^3).

erad *0 ir*

Radiation energy density at node *ir* (erg/cm^3).

etot *0 ir*

Total energy density (material + radiation) at node *ir* (erg/cm^3).

cv_e *0 ir*

Electron specific heat (constant volume) at node *ir* (erg/cm^3/K).

cv_i *0 ir*

Ion specific heat (constant volume) at node *ir* (erg/cm^3/K).

pmat *0 ir*

Material pressure at node *ir* (erg/cm^3).

pmat_e *0 ir*

Electron pressure at node *ir* (erg/cm^3).

pmat_i *0 ir*

Ion pressure at node *ir* (erg/cm^3).

prad *0 ir*

Radiation pressure at node *ir* (erg/cm^3).

ptot *0 ir*

Total pressure (material + radiation) at node *ir* (erg/cm^3).

abar *0 ir*

Average ion atomic weight at node *ir*.

zbar *0 ir*

Average ion charge at node *ir*.

z2bar *0 ir*

Average of square of ion charge at node *ir*.

qmoment_*n* *iz ir*

The n^{th} moment of the charge distribution of element *iz* at node *ir*.

cei *0 ir*

Electron-ion coupling coefficient at node *ir* (cm^3/sec).

coulog *0 ir*

Coulomb logarithm at node *ir*.

gplasma *0 ir*

Plasma coupling constant at node *ir*.

bx *0 ir*

Component of magnetic field in x- (or r-) direction at node *ir* (G).

by *0 ir*

Component of magnetic field in y- (or z-) direction at node *ir* (G).

bz *0 ir*

Component of magnetic field in z- (or phi-) direction at node *ir* (G).

bfield *0 ir*

Magnetic field strength at node *ir* (G).

Populations:

level *iz 0 iso i*

Index of state (*iso*,*i*) of element *iz*. If *iso* is zero, the index is with respect to the entire model, otherwise, the index is with respect to isoelectronic sequence *iso*.

iso *iz 0 iso i*

Isoelectronic sequence of state (*iso*,*i*) of element *iz*. When used as an **xvar**, *iso* and *i* will be ignored.

npq *iz 0 iso i*

Principal quantum number of state (*iso*,*i*) of element *iz*.

nlq *iz 0 iso i*

Angular momentum quantum number of state (*iso*,*i*) of element *iz*.

npqmax *iz 0 iso i*

Maximum principal quantum number of state (*iso*,*i*) of element *iz* before lines merge (Inglis-Teller limit).

shell_*xx* *iz 0 iso i*

Shell occupation for state (*iso*,*i*) of element *iz*, where ***xx*** is an index denoting the shell (1=k-shell, 2=l-shell, 3=m-shell, …).

y *iz ir iso i*

Ion density of state (*iso*,*i*) of element *iz* at node *ir* (1/cm^3).

yfrac *iz ir iso i*

Fractional population of state (*iso*,*i*) of element *iz* at node *ir*.

yiso *iz ir iso*

Ion density of isoelectronic sequence *iso* of element *iz* at node *ir* (1/cm^3).

yisofrac *iz ir iso*

Fractional population of isoelectronic sequence *iso* of element *iz* at node *ir*.

y1 *iz ir iso*

Ion density of isoelectronic sequence *iso* of element *iz* at node *ir* (1/cm^3).

y1frac *iz ir iso*

Fractional population of isoelectronic sequence *iso* of element *iz* at node *ir*.

ytot *iz ir*

Total ion density of element *iz* at node *ir* (1/cm^3).

gammatot *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* (1/s).

gamma *iz ir iso i*

Destruction rate of state (*iso*,*i*) of element *iz* at node *ir* used in calculation of line widths (1/s).

nbound *iz ir*

Average number of bound electrons for ions of element *iz* at node *ir*.

isomin *iz ir*

Minimum isoelectronic sequence included for ions of element *iz* at node *ir*.

isomax *iz ir*

Maximum isoelectronic sequence included for ions of element *iz* at node *ir*.

elev0 *iz ir iso i*

Energy of state (*iso*,*i*) of element *iz* at node *ir*, relative to zero energy for the ground state of the lowest isoelectronic sequence with no continuum lowering (eV).

elev *iz ir iso i*

Energy of state (*iso*,*i*) of element *iz* at node *ir*, relative to zero energy for the ground state of the lowest isoelectronic including effects of continuum lowering (eV).

glev0 *iz ir iso I*

Statistical weight of state (*iso*,*I*) of element *iz* at node *ir* with no continuum lowering.

glev *iz ir iso i*

Statistical weight of state (*iso*,*i*) of element *iz* at node *ir* including effects of continuum lowering.

plev *iz ir iso i*

Statistical weight multiplier (degeneracy lowering) of state (*iso*,*i*) of element *iz* at node *ir*.

delev *iz ir iso i*

Energy lowering required for state (*iso*,*i*) of element *iz* at node *ir* to vanish (eV).

elower *iz ir iso i*

Energy lowering of state (*iso*,*i*) of element *iz* at node *ir* (eV).

eint0 *iz ir*

Internal energy of element *iz* at node *ir* with no continuum lowering (erg/cm^3).

deint0dt *iz ir*

Derivative w.r.t. electron temperature of internal energy of element *iz* at node *ir* with no continuum lowering (erg/cm^3/eV).

eint *iz ir*

Internal energy of element *iz* at node *ir* including effects of continuum lowering (erg/cm^3).

deintdt *iz ir*

Derivative w.r.t. electron temperature of internal energy of element *iz* at node *ir* including effects of continuum lowering (erg/cm^3/eV).

pfnc0 *iz ir*

Partition function of element *iz* at node *ir* with no continuum lowering.

pfnc *iz ir*

Partition function of element *iz* at node *ir* including effects of continuum lowering.

Rates:

(+) signifies the transition from lower to upper state(s)
(-) signifies the transition from upper to lower state(s)

Edits specific to a particular transition *itr* and node *ir* will return a zero value if that transition has not been calculated for that node. This will occur if the populations involved are not included for that node.

acoeff *itr ir*

Einstein A-value (spontaneous de-excitation rate) for photoexcitation transition *itr* (1/s). Although this quantity is independent of position and plasma conditions, a valid node index *ir* must be present

avoigt0 *itr ir*

Voigt parameter for photoexcitation transition *itr* at node *ir*.

uvoigt0 *itr ir*

Maximum value of Voigt function for photoexcitation transition *itr* at node *ir*.

gain *itr ir*

Gain for photoexcitation transition *itr* at node *ir* (1/cm).

widthn *itr ir*

Natural width for photoexcitation transition *itr* at node *ir* (eV).

widths *itr ir*

Stark width for photoexcitation transition *itr* at node *ir* (eV).

widthd *itr ir*

Doppler width for photoexcitation transition *itr* at node *ir* (eV).

pump *itr ir*

Photon intensity for photoexcitation transition *itr* at node *ir* (photons/mode).

escprob *itr ir*

Escape probability for photoexcitation transition *itr* at node *ir*.

bsplit *itr ir*

Zeeman splitting for photoexcitation transition *itr* at node *ir* (eV).

c_rate+ *itr ir*
c_rate- *itr ir*

Rate for collisional transition *itr* at node *ir* (1/sec).

p_rate+ *itr ir*
p_rate- *itr ir*

Rate for photo transition *itr* at node *ir* (1/sec).

a_rate+ *itr ir*
a_rate- *itr ir*

Rate for auger transition *itr* at node *ir* (1/sec).

d_rate+ *itr ir*
d_rate- *itr ir*

Rate for ion collisional transition *itr* at node *ir* (1/sec).

c_tot+ *itr ir*
c_tot- *itr ir*
c_net *itr ir*

Total rate for collisional transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (1/sec).

p_tot+ *itr ir*
p_tot- *itr ir*
p_net *itr ir*

Total rate for photo transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (1/sec).

a_tot+ *itr ir*
a_tot- *itr ir*
a_net *itr ir*

Total rate for auger transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (1/sec).

d_tot+ *itr ir*
d_tot- *itr ir*
d_net *itr ir*

Total rate for ion collisional transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (1/sec).

ec_rate+ *itr ir*

ec_rate- *itr ir*

Energy contribution to free electrons for collisional transition *itr* at node *ir* (eV).

ep_rate+ *itr ir*
ep_rate- *itr ir*

Energy contribution to free electrons for photo transition *itr* at node *ir* (eV).

ea_rate+ *itr ir*
ea_rate- *itr ir*

Energy contribution to free electrons for auger transition *itr* at node *ir* (eV).

ed_rate+ *itr ir*
ed_rate- *itr ir*

Energy contribution to free electrons for ion collisional transition *itr* at node *ir* (eV).

ce_rate+ *itr ir*
ce_rate- *itr ir*

Energy deposition rate for collisional transition *itr* at node *ir* (eV/sec).

pe_rate+ *itr ir*
pe_rate- *itr ir*

Energy deposition rate for photo transition *itr* at node *ir* (eV/sec).

ae_rate+ *itr ir*
ae_rate- *itr ir*

Energy deposition rate for auger transition *itr* at node *ir* (eV/sec).

de_rate+ *itr ir*
de_rate- *itr ir*

Energy deposition rate for ion collisional transition *itr* at node *ir* (eV/sec).

rp_rate+ *itr ir*
rp_rate- *itr ir*

Energy contribution to radiation field for photo transition *itr* at node *ir* (eV).

ra_rate+ *itr ir*
ra_rate- *itr ir*

Energy contribution to radiation field for auger transition *itr* at node *ir* (eV).

pr_rate+ *itr ir*
pr_rate- *itr ir*

Energy contribution rate to radiation field for photo transition *itr* at node *ir* (eV/sec).

ar_rate+ *itr ir*
ar_rate- *itr ir*

Energy contribution rate to radiation field for auger transition *itr* at node *ir* (eV/sec).

ce_tot+ *itr ir*
ce_tot- *itr ir*
ce_net *itr ir*

Total energy deposition rate for collisional transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (eV/sec).

pe_tot+ *itr ir*
pe_tot- *itr ir*
pe_net *itr ir*

Total energy deposition rate for photo transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (eV/sec).

ae_tot+ *itr ir*
ae_tot- *itr i*
ae_net *itr ir*

Total energy deposition rate for auger transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (eV/sec).

de_tot+ *itr ir*
de_tot- *itr ir*
de_net *itr ir*

Total energy deposition rate for ion collisional transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (eV/sec).

pr_tot+ *itr ir*
pr_tot- *itr ir*
pr_net *itr ir*

Total energy deposited into radiation field for photo transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (eV/sec).

ar_tot+ *itr ir*
ar_tot- *itr ir*
ar_net *itr ir*

Total energy deposited into radiation field for auger transition *itr* at node *ir*, averaged over initial state(s) and summed over final state(s) (eV/sec).

gamma_cx *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to collisional excitation processes (1/s).

gamma_ci *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to collisional ionization processes (1/s).

gamma_c *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to all collisional processes (1/s).

gamma_px *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to photoexcitation processes (1/s).

gamma_pi *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to photoionization processes (1/s).

gamma_p *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to all photo processes (1/s).

gamma_a *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to all auger processes (1/s).

gamma_dx *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to ion collisional excitation processes (1/s).

gamma_di *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to ion collisional ionization processes (1/s).

gamma_d *iz ir iso i*

Total destruction rate of state (*iso*,*i*) of element *iz* at node *ir* due to all ion collisional processes (1/s).

eigenval *iz ir 0 i*

i^{th} eigenvalue of the rate matrix for element *iz* at node *ir*. Requesting this edit is the same as requesting both **eigenv_r** and **eigenv_i**. The eigenvalues are sorted in descending order of the real part of the eigenvalue.

eigenv_r *iz ir 0 i*

Real part of the i^{th} eigenvalue of the rate matrix for element *iz* at node *ir*. The eigenvalues are sorted in descending order of the real part of the eigenvalue.

eigenv_i *iz ir 0 i*

Imaginary part of the i^{th} eigenvalue of the rate matrix for element *iz* at node *ir*. The eigenvalues are sorted in descending order of the real part of the eigenvalue.

Heating rates:

x = "a" signifies auger processes
x = "c" signifies electron collisional processes
x = "p" signifies photo processes

heatb *0 ir*

Free electron heating rate due to bremsstrahlung absorption at node *ir* (eV/sec).

heat*x* *iz ir*

Free electron heating rate due to atomic processes of type ***x*** from element *iz* at node *ir* (eV/sec).

heati *iz ir*

Ion heating rate due to ion collisional processes from element *iz* at node *ir* (eV/sec).

coolb *0 ir*

Free electron cooling rate due to bremsstrahlung emission at node *ir* (eV/sec).

cool*x* *iz ir*

Free electron cooling rate due to atomic processes of type ***x*** from element *iz* at node *ir* (eV/sec).

cooli *iz ir*

Ion cooling rate due to ion collisional processes from element *iz* at node *ir* (eV/sec).

heatbt *0 ir*

Net free electron heating rate due to bremsstrahlung absorption and emission at node *ir* (eV/sec).

heat*x*t *iz ir*

Net free electron heating rate due to atomic processes of type ***x*** from element *iz* at node *ir* (eV/sec).

heatit *iz ir*

Net ion heating rate due to ion collisional processes from element *iz* at node *ir* (eV/sec).

heatt *0 ir*

Net free electron heating rate due to all processes at node *ir* (eV/sec).

heatti *0 ir*

Net ion heating rate due to all processes at node *ir* (eV/sec).

dhtbdt *0 ir*

Derivative w.r.t. electron temperature of free electron heating rate due to bremsstrahlung absorption at node *ir* (1/sec).

dht*x*dt *iz ir*

Derivative w.r.t. electron temperature of free electron heating rate due to atomic processes of type ***x*** from element *iz* at node *ir* (1/sec).

dhtidt *iz ir*

Derivative w.r.t. ion temperature of ion heating rate due to ion collisional processes from element *iz* at node *ir* (1/sec).

dclbdt *0 ir*

Derivative w.r.t. electron temperature of free electron cooling rate due to bremsstrahlung emission at node *ir* (1/sec).

dcl*x*dt *iz ir*

Derivative w.r.t. electron temperature of free electron cooling rate due to atomic processes of type ***x*** from element *iz* at node *ir* (1/sec).

dclidt *iz ir*

Derivative w.r.t. ion temperature of ion cooling rate due to ion collisional processes from element *iz* at node *ir* (1/sec).

dht*x*tdt *iz ir*

Derivative w.r.t. electron temperature of net free electron heating rate due to atomic processes of type ***x*** from element *iz* at node *ir* (1/sec).

dhtitdt *iz ir*

Derivative w.r.t. ion temperature of net ion heating rate due to ion collisional processes from element *iz* at node *ir* (1/sec).

dhttdt *0 ir*

Derivative w.r.t. electron temperature of net free electron heating rate due to all processes at node *ir* (1/sec).

econd *0 ir*

Electron thermal conduction coefficient at node *ir* (cm^2/s).

esrce *0 ir*

Energy sources into free electrons at node *ir* ($erg/cm^3/s$).

esrcev *0 ir*

Free electron heating rate due to energy sources at node *ir* (eV/s).

esrci *0 ir*

Energy sources into ions at node *ir* ($erg/cm^3/s$).

esrciv *0 ir*

Ion heating rate due to energy sources at node *ir* (eV/s).

desrce *0 ir*

Derivative of free electron heating rate at node *ir* w.r.t. temperature (1/s).

desrci *0 ir*

Derivative of ion heating rate at node *ir* w.r.t. temperature (1/s).

Continuum radiation:

(+) signifies the direction of increasing ir in 1-d geometries
(-) signifies the direction of decreasing ir in 1-d geometries
(+,-) are equivalent in 2-d geometries

ifr *0 0 igp*

Continuum group index.

nf

Number of continuum groups.

energy *0 0 igp*

Energy at center of continuum group *igp* (eV).

ebins *0 0 igp*

Energy at boundaries of continuum group *igp* (eV). Using this as an x-variable results in a histogram, with y-variable values evaluated at group centers.

freq *0 0 igp*

Frequency at center of continuum group *igp* (Hz).

fbins *0 0 igp*

Frequency at boundaries of continuum group *igp* (Hz). Using this as an x-variable results in a histogram, with y-variable values evaluated at group centers.

nu *0 0 igp*

Frequency at center of continuum group *igp* (Hz).

wvl *0 0 igp*

Wavelength at center of continuum group *igp* (cm).

wbins *0 0 igp*

Wavelength at boundaries of continuum group *igp* (cm). Using this as an x-variable results in a histogram, with y-variable values evaluated at group centers.

nray

Number of directions used in transfer calculation.

neray

Number of additional directions used in transfer calculation to satisfy edit requests.

ndtct

Number of detectors currently defined for transfer calculation.

ckap *0 ir igp*

Absorption coefficient at node *ir* for group *igp* (1/cm).

cemis *0 ir igp*

Emission coefficient at node *ir* for group *igp* (erg/cm^3/sec/Hz).

scat *0 ir igp*

Scattering coefficient at node *ir* for group *igp* (1/cm).

nescat *0 ir*

Effective electron density for scattering at node *ir* (1/cm^3).

bnu *0 ir igp*

Black-body radiation intensity appropriate to the electron temperature at node *ir* for group *igp* (erg/cm^2/sec/Hz).

bnur *0 ir igp*

Black-body radiation intensity appropriate to the radiation temperature at node *ir* for group *igp* (erg/cm^2/sec/Hz).

jbndry *ix 0 igp*

Radiation intensity from xfile *ix* for group *igp* (erg/cm^2/sec/Hz).

jnu *0 ir igp*

Radiation intensity (angle-averaged) at node *ir* for group *igp* (erg/cm^2/sec/Hz).

jnul *0 ir igp*

Radiation intensity (angle-averaged) from lines at node *ir* for group *igp* (erg/cm^2/sec/Hz).

jnutot *0 ir igp*

Radiation intensity (angle-averaged) from continuum and lines at node *ir* for group *igp* (erg/cm^2/sec/Hz).

jrate *0 ir igp*

Radiation intensity (angle-averaged) used for atomic kinetics at node *ir* for group *igp* (erg/cm^2/sec/Hz).

jratei *0 ir igp*

Radiation intensity (angle-averaged) used for photoionizations at node *ir* for group *igp* (erg/cm^2/sec/Hz).

jratex *0 ir igp*

Radiation intensity (angle-averaged) used for photoexcitations at node *ir* for group *igp* (erg/cm^2/sec/Hz).

eradc *0 ir*

Continuum radiation energy density at node *ir* (erg/cm^3).

eradl *0 ir*

Line radiation energy density at node *ir* (erg/cm^3).

erad *0 ir*

Total radiation energy density (from continuum and lines) at node *ir* (erg/cm^3).

The following three sets of edits specify optical depths. In 1-d geometries, the extent is from node 1 to node *ir*, or for all nodes if *ir*=0. In 2-d geometries, the extent is to node *ir* along the appropriate k-line (for **tau_k**) or l-line (for **tau_l**).

taukap *0 ir igp*
taukapk *0 ir igp*
taukapl *0 ir igp*

Absorption optical depth for group *igp*. from node 1 to node *ir* or total optical depth (for all nodes) if *ir*=0 (for 1-d geometries only).

tauscat *0 ir igp*
tauscatk *0 ir igp*
tauscatl *0 ir igp*

Scattering optical depth for group *igp*.

tautot *0 ir igp*
tautotk *0 ir igp*
tautotl *0 ir igp*

Total optical depth for group *igp*.

inu+ *0 ir igp idir*
inu- *0 ir igp idir*

Specific intensity at node *ir* for group *igp* in direction *idir* in (+) or (-) direction (erg/cm^2/sec/Hz/ster).

cflux+ *0 ir igp*

cflux- *0 ir igp*

Flux at node *ir* for group *igp* in (+) or (-) direction (erg/cm^2/sec/Hz) (for 1-d geometries only).

cfluxx+ *0 ir igp*
cfluxx- *0 ir igp*

Flux at node *ir* for group *igp* in (+) or (-) x-direction or r-direction (erg/cm^2/sec/Hz) (for 2-d geometries only).

cfluxy+ *0 ir igp*
cfluxy- *0 ir igp*

Flux at node *ir* for group *igp* in (+) or (-) y-direction or z-direction (erg/cm^2/sec/Hz) (for 2-d geometries only).

eflux+ *0 ir*
eflux- *0 ir*

Integrated flux at node *ir* in (+) or (-) direction (erg/cm^2/sec) (for 1-d geometries only).

efluxx+ *0 ir*
efluxx- *0 ir*

Integrated flux at node *ir* in (+) or (-) x-direction or r-direction (erg/cm^2/sec) (for 2-d geometries only).

efluxy+ *0 ir*
efluxy- *0 ir*

Integrated flux at node *ir* in (+) or (-) y-direction or z-direction (erg/cm^2/sec) (for 2-d geometries only).

cdetect *0 idtct igp*

Flux for group *igp* seen by detector *idtct* (erg/cm^2/sec/Hz) (for 2-d geometries only).

edetect *0 idtct igp*

Integrated flux seen by detector *idtct* (erg/cm^2/sec) (for 2-d geometries only).

tauray *0 ir igp idir*

In 1-d geometries, total optical depth from node 1 to node *ir* for group *igp*, or total optical depth (for all nodes) if *ir*=0, in direction *idir*.

In 2-d geometries, total optical depth from node *ir* to boundary for group *igp* in direction *idir*.

Spectral radiation:

(+) signifies the direction of increasing ir in 1-d geometries
(-) signifies the direction of decreasing ir in 1-d geometries
(+,-) are equivalent in 2-d geometries

Two indices are required to specify a particular energy. The spectral range index *is* specifies the energy range, corresponding to a **spectral-group** command. The energy index *isp* specifies the energy within that range. The value *is*=0 corresponds to the full spectrum.

isp *is 0 isp*

Spectral group index within the range specified by the index *is*.

nsp *is*

Number of spectral groups within the range specified by the index *is*.

sp_energy *is 0 isp*

Energy for spectral group (*is,isp*) (eV).

sp_freq *is 0 isp*

Frequency for spectral group (*is,isp*) (Hz).

sp_nu *is 0 isp*

Frequency for spectral group (*is,isp*) (Hz).

sp_wvl *is 0 isp*

Wavelength for spectral group (*is,isp*) (cm).

emis_sp *is ir isp*

Emission coefficient at node *ir* for spectral group (*is,isp*) without doppler shifts (erg/cm^3/sec/Hz).

emis_sp+ *is ir isp idir*
emis_sp- *is ir isp idir*

Emission coefficient at node *ir* for spectral group (*is,isp*) in direction *idir* in (+) or (-) direction (erg/cm^3/sec/Hz).

kappa_sp *is ir isp*

Absorption coefficient at node *ir* for spectral group (*is,isp*) without doppler shifts (1/cm).

kappa_sp+ *is ir isp idir*
kappa_sp- *is ir isp idir*

Absorption coefficient at node *ir* for spectral group (*is,isp*) in direction *idir* in (+) or (-) direction (1/cm).

jsp *is ir isp*

Spectral radiation intensity (angle-averaged) at node *ir* for spectral group (*is,isp*) (erg/cm^2/sec/Hz).

isp+ *is ir isp idir*
isp- *is ir isp idir*

Spectral specific intensity at node *ir* for spectral group (*is,isp*) in direction *idir* in (+) or (-) direction (erg/cm^2/sec/Hz/ster).

spflux+ *is ir isp*
spflux- *is ir isp*

Flux at node *ir* for spectral group (*is,isp*) in (+) or (-) direction (erg/cm^2/sec/Hz) (for 1-d geometries only).

spfluxx+ *is ir isp*
spfluxx- *is ir isp*

Flux at node *ir* for spectral group (*is,isp*) in (+) or (-) x-direction or r-direction (erg/cm^2/sec/Hz) (for 2-d geometries only).

spfluxy+ *is ir isp*
spfluxy- *is ir isp*

Flux at node *ir* for spectral group (*is,isp*) in (+) or (-) y-direction or z-direction (erg/cm^2/sec/Hz) (for 2-d geometries only).

speflux+ *is ir*
speflux- *is ir*

Integrated flux at node *ir* in (+) or (-) direction for spectral range *is* (erg/cm^2/sec) (for 1-d geometries only).

spefluxx+ *is ir*
spefluxx- *is ir*

Integrated flux at node *ir* in (+) or (-) x-direction or r-direction for spectral range *is* (erg/cm^2/sec) (for 2-d geometries only).

spefluxy+ *is ir*
spefluxy- *is ir*

Integrated flux at node *ir* in (+) or (-) y-direction or z-direction for spectral range *is* (erg/cm^2/sec) (for 2-d geometries only).

spdetect *is idtct isp*

Flux for spectral group *isp* seen by detector *idtct* (erg/cm^2/sec/Hz) (for 2-d geometries only).

tausp *is ir isp*
tauspk *is ir isp*
tauspl *is ir isp*

Optical depth for spectral group (*is,isp*). In 1-d geometries, the extent is from node 1 to node *ir*, or for all nodes if *ir*=0. In 2-d geometries, the extent is to node *ir* along the appropriate k-line (for **tauspk**) or l-line (for **tausp**).

tausp+ *is ir isp idir*
tausp- *is ir isp idir*

In 1-d geometries, optical depth from node 1 to node *ir* for spectral group (*is,isp*), or total optical depth (for

all nodes) if *ir*=0, in direction *idir* in (+) or (-) direction.

In 2-d geometries, optical depth from node *ir* to boundary for spectral group (*is*,*isp*) in direction *idir*.

jsparea *is ir isp*

For spherical geometry, the area-integrated intensity for spectral group (*is*,*isp*) emergent from a circular mask centered on r=0 and extending to r=r(*ir*) (erg/sec/Hz/ster).

For cylindrical geometry, the area-integrated intensity for spectral group (*is*,*isp*) emergent from a mask of length param(84) centered on r=0 and extending to r=r(ir), viewed from an angle θ with respect to the z-axis given by
$\theta = \cos^{-1}(\text{param}(83))$ (erg/sec/Hz/ster).

Line radiation:

(+) signifies the direction of increasing ir in 1-d geometries
(-) signifies the direction of decreasing ir in 1-d geometries
(+,-) are equivalent in 2-d geometries

avoigt *iline ir*

Voigt parameter for line *iline* at node *ir*.

uvoigt *iline ir*

Maximum value of Voigt function for line *iline* at node *ir*.

dopwidth *iline ir*

Doppler width of line *iline* at node *ir* (eV).

eline *iline ir*

Energy density in radiation of line *iline* at node *ir* (erg/cm^3).

jmode *iline*

Conversion factor from (photons/mode) to (erg/cm^2/sec/Hz) for line *iline*.

jbar *iline ir*

Line strength of line *iline* at node *ir* (photons/mode).

jbarcgs *iline ir*

Line strength of line *iline* at node *ir* (erg/cm^2/sec/Hz).

jbarx *iline ir*

Intermediate value of line strength of line *iline* at node *ir* (photons/mode). This quantity may only be edited as a snapshot over the entire mesh and one edit will appear for each iteration during a timestep.

lineiter *iline*

Number of iterations taken by the approximate operator formalism for line *iline* during a timestep. This quantity may only be edited vs. time.

jsat *iline ir*

Saturation intensity for line *iline* at node *ir* (photons/mode).

jsatcgs *iline ir*

Saturation intensity for line *iline* at node *ir* (erg/cm^2/sec/Hz).

nrb *iline ir*

Net radiative bracket for line *iline* at node *ir*.

sigma *iline ir*

Integrated absorption cross-section for line *iline* at node *ir* (cm^2).

ylinel *iline ir*

Effective lower level population for line *iline* at node *ir*, i.e. $y_l - (g_l/g_u)y_u$ (1/cm^3).

ylineu *iline ir*

Effective upper level population for line *iline* at node *ir*, i.e. $(g_l/g_u)y_u$ (1/cm^3).

dyldj *iline ir*

Derivative of logarithm of effective lower level population w.r.t. jbar for line *iline* at node *ir* (modes/photon).

dyudj *iline ir*

Derivative of logarithm of effective upper level population w.r.t. jbar for line *iline* at node *ir* (modes/photon).

prdfrac *iline ir*

Fraction of emissivity which is redistributed for prd line *iline* at node *ir*.

ifrline *iline 0 ifr*

Line group index for line *iline*.

evline *iline 0 ifr*

Energy of bin *ifr* in line *iline* w.r.t. center of line *iline* (eV).

nuline *iline 0 ifr*

Frequency of bin *ifr* in line *iline* w.r.t. center of line *iline* (Hz).

jline *iline ir ifr*

Average intensity (in fluid frame) in line *iline* at node *ir* for bin *ifr* (photons/mode).

jline0 *iline ir ifr*

Average intensity (in lab frame) in line *iline* at node *ir* for bin *ifr* (photons/mode).

jlinecgs *iline ir ifr*

Average intensity (in fluid frame) in line *iline* at node *ir* for bin *ifr* (erg/cm^2/sec/Hz/ster).

jline0cgs *iline ir ifr*

Average intensity (in lab frame) in line *iline* at node *ir* for bin *ifr* (erg/cm^2/sec/Hz/ster).

lemis *iline ir ifr*

Emission coefficient for line *iline* at node *ir* for bin *ifr* (photons/mode/cm).

lemistot *iline ir ifr*

Total emission coefficient (lines + continuum) in line *iline* at node *ir* for bin *ifr* (photons/mode/cm).

lkap *iline ir ifr*

Absorption coefficient for line *iline* at node *ir* for bin *ifr* (1/cm).

lkaptot *iline ir ifr*

Total absorption coefficient (lines + continuum) in line *iline* at node *ir* for bin *ifr* (1/cm).

srcline *iline ir ifr*

Source function in line *iline* at node *ir* for bin *ifr* (photons/mode).

tauline *iline ir ifr*
taulinek *iline ir ifr*
taulinel *iline ir ifr*

Optical depth in line *iline* for bin *ifr*. In 1-d geometries, the extent is from node 1 to node *ir*, or for all nodes if *ir*=0. In 2-d geometries, the extent is to node *ir* along the appropriate k-line (for **taulinek**) or l-line (for **taulinel**).

tauline+ *iline ir ifr idir*
tauline- *iline ir ifr idir*

In 1-d geometries, optical depth in line *iline* from node 1 to node *ir* for bin *ifr*, or total optical depth (for all nodes) if *ir*=0, in direction *idir* in (+) or (-) direction.

In 2-d geometries, optical depth from node *ir* to boundary for bin *ifr* in direction *idir*.

iline+ *iline ir ifr idir*
iline- *iline ir ifr idir*

Specific intensity in line *iline* at node *ir* for bin *ifr* in direction *idir* in (+) or (-) direction (photons/mode).

iflux+ *iline ir ifr*
iflux- *iline ir ifr*

Flux in line *iline* at node *ir* for bin *ifr* in (+) or (-) direction (photons/mode) (for 1-d geometries only).

lflux+ *iline ir*
lflux- *iline ir*

Integrated flux in line *iline* at node *ir* in (+) or (-) direction (erg/cm^2/sec) (for 1-d geometries only).

ifluxx+ *iline ir ifr*
ifluxx- iline *ir ifr*

Flux in line *iline* at node *ir* for bin *ifr* in (+) or (-) x-direction or r-direction (photons/mode) (for 2-d geometries only).

ifluxy+ *iline ir ifr*
ifluxy- iline *ir ifr*

Flux in line *iline* at node *ir* for bin *ifr* in (+) or (-) y-direction or z-direction (photons/mode) (for 2-d geometries only).

lfluxx+ *iline ir*
lfluxx- iline *ir*

Integrated flux in line *iline* at node *ir* in (+) or (-) x-direction or r-direction (erg/cm^2/sec) (for 2-d geometries only).

lfluxy+ *iline ir*
lfluxy- iline *ir*

Integrated flux in line *iline* at node *ir* in (+) or (-) y-direction or z-direction (erg/cm^2/sec) (for 2-d geometries only).

idetect *iline idtct ifr*

Flux in line *iline* for bin *ifr* seen by detector *idtct* (erg/cm^2/sec/Hz) (for 2-d geometries only).

ldetect *iline idtct*

Integrated flux in line *iline* seen by detector *idtct* (erg/cm^2/sec) (for 2-d geometries only).

lemis+ *iline ir ifr idiry*
lemis- *iline ir ifr idir*

Emission coefficient for line *iline* at node *ir* for bin *ifr* in direction *idir* in (+) or (-) direction (photons/mode/cm).

lemist+ *iline ir ifr idir*
lemist- *iline ir ifr idir*

Total emission coefficient (lines + continuum) in line *iline* at node *ir* for bin *ifr* in direction *idir* in (+) or (-) direction (photons/mode/cm)

lkap+ *iline ir ifr idir*
lkap- *iline ir ifr idir*

Absorption coefficient for line *iline* at node *ir* for bin *ifr* in direction *idir* in (+) or (-) direction (1/cm).

lkapt+ *iline ir ifr idir*
lkapt- *iline ir ifr idir*

Total absorption coefficient (lines + continuum) in line *iline* at node *ir* for bin *ifr* in direction *idir* in (+) or (-) direction (1/cm).

srcline+ *iline ir ifr idir*
srcline- *iline ir ifr idir*

Source function for line *iline* at node *ir* for bin *ifr* in direction *idir* in (+) or (-) direction (photons/mode).

Convergence:

ntry

Number of iterations for current timestep.

diffy *iz ir iso i*

Fractional change in population of state (*iso*,*i*) of element *iz* at node *ir* for current timestep.

diffy1 *iz ir iso*

Fractional change in population of isoelectronic sequence *iso* of element *iz* at node *ir* for current timestep.

diffte *0 ir*

Fractional change in electron temperature at node *ir* for current timestep.

diffti *0 ir*

Fractional change in ion temperature at node *ir* for current timestep.

diffj *iline ir*

Fractional change in line strength (jbar) for line *iline* at node *ir* for current timestep.

diffymax

Maximum fractional change in isosequence populations for current timestep.

difftmax

Maximum fractional change in electron or ion temperature for current timestep.

diffjmax

Maximum fractional change in line strengths for current timestep.

ndifft

Zone with maximum fractional change in electron or ion temperature for current timestep.

ndiffy

Zone with maximum fractional change in isosequence populations for current timestep.

ndiff_iz

Model index with maximum fractional change in isosequence populations for current timestep.

ndiff_iso

Isosequence with maximum fractional change in isosequence populations for current timestep.

ndiffj

Zone with maximum fractional change in line strengths for current timestep.

ndiff_line

Line index with maximum fractional change in line strengths for current timestep.

iter_temp

Number of iterations for 2d/3d thermal conduction linear solver.

err_temp

Residual magnitude for 2d/3d thermal conduction linear solver.

Sample Generator Files

0-d kinetics:

```
c               **** 0-d kinetics/temperature test ****

c -------------------------------------------------------------
c   Materials
c -------------------------------------------------------------

atoms dnl06ha.dat  6. 12.
atoms dnl26h1.dat 26. 55.85

region  1 1  1000.
  element  2  1.00e20  20 26
  element  1  3.00e21


c -------------------------------------------------------------
c   Sources
c -------------------------------------------------------------

xfile x0dty.dat

xmult 1 pbins 0.1


c -------------------------------------------------------------
c   Controls
c -------------------------------------------------------------

tstart 0.
tquit  1.0e-8

restart


c -------------------------------------------------------------
c   Switches and Parameters
c -------------------------------------------------------------

switch  11   1            ! make .plt file
switch  25   1            ! time-dependent
switch  28   0            ! do LTE initialization
switch  29   1            ! use fixed timesteps
switch  30  10            ! dump every n timesteps
switch  31   1            ! do temperature calculation
switch  44   3            ! maximum # of iterations

param   41  1.e-10        ! initial timestep
param   44  2.e-11        ! minimum timestep
param   45  2.e-10        ! maximum timestep
param   48  0.1           ! electron-ion coupling multiplier
param   61  1.e-4         ! iso-sequence population threshold


c -------------------------------------------------------------
c   Edits
c -------------------------------------------------------------

transition 1 2 10 3 10 6
```

```
transition 2 2 10 3 10 7
transition 3 2 10 3 10 8
transition 4 2 10 3 10 9

plot "Y1FRAC vs ISO (IRON)"
  xvar  iso     2
  yvar  y1frac  2 1

plot "TEV, TIV vs TIME"
  xvar  time
  yvar  tev    0 1
  yvar  tiv    0 1

plot "NE vs TIME"
  xvar  time
  yvar  ne     0 1

plot "Y1FRAC (IRON) vs TIME"
  xvar  time
  yvar  y1frac  2 1 26:0:-1

plot "ISOMIN, ISOMAX (CARBON) vs TIME"
  xvar  time
  yvar  isomin 1 1
  yvar  isomax 1 1

plot "ISOMIN, ISOMAX (IRON) vs TIME"
  xvar  time
  yvar  isomin 2 1
  yvar  isomax 2 1

plot "HEATC, COOLC vs TIME"
  xvar  time
  yvar  heatc   2 1
  yvar  coolc   2 1
  yvar  heatct  2 1

plot "HEATP, COOLP vs TIME"
  xvar  time
  yvar  heatp   2 1
  yvar  coolp   2 1
  yvar  heatpt  2 1

plot "HEATA, COOLA vs TIME"
  xvar  time
  yvar  heata   2 1
  yvar  coola   2 1
  yvar  heatat  2 1

plot "HEATB, COOLB vs TIME"
  xvar  time
  yvar  heatb   0 1
  yvar  coolb   0 1
  yvar  heatbt  0 1

plot "DHEATx vs TIME"
  xvar  time
  yvar  dhtcdt  2 1
  yvar  dhtpdt  2 1
  yvar  dhtadt  2 1
  yvar  dhtbdt  0 1

plot "EINT vs TIME"
  xvar  time
  yvar  eint    1 1
  yvar  eint    2 1
```

```
plot "DEINTDT vs TIME"
  xvar  time
  yvar  deintdt 1 1
  yvar  deintdt 2 1

plot "EIGENV_R (CARBON) vs TIME"
  xvar  time
  yvar  eigenv_r  1 1 0 1:5

plot "EIGENV_R (IRON) vs TIME"
  xvar  time
  yvar  eigenv_r  2 1 0 1:5
```

1-d radiation transfer

```
c    **** steady-state spherical continuum-lowering spectrum calculation ****

c -------------------------------------------------------------
c   Aliases
c -------------------------------------------------------------

alias n0   1
alias n1  14

alias ir1  1
alias ir2  4
alias ir3  7
alias ir4 11
alias ir5 14


c -------------------------------------------------------------
c   Materials
c -------------------------------------------------------------

atoms rtdat18.dat 18. 40.

region  n0 n1  1000.
  element 1 6.11e22  1 5


c -------------------------------------------------------------
c   Geometry
c -------------------------------------------------------------

geometry sphere

rlin n0 n1 0.000 0.005


c -------------------------------------------------------------
c   Radiation
c -------------------------------------------------------------

angles 2

ebins 100 5. 10000. 1.05

spectrum 40     5.  1000.  1.
spectrum 20  1000.  2800.  1.
spectrum 40  2800.  4200.  1.
spectrum 20  4200. 10000.  1.

spectral-group 10 1000. 2800.
spectral-group 20 2800. 4200.
spectral-group 30 4200. 5000.


linedefault approximate

line 1 1 1 1 1 2
  lbins 15 50.0 1.1

line 2 1 1 1 1 3
  lbins 15 15.0 1.1
```

```
c -------------------------------------------------------------
c   Controls
c -------------------------------------------------------------

tstart 0.
tquit  5.0e-10

restart


c -------------------------------------------------------------
c   Switches and Parameters
c -------------------------------------------------------------

switch  11   1            ! make .plt file
switch  29   2            ! use variable timesteps
switch  30  10            ! dump every n timesteps
switch  31   0            ! no temperature calculation
switch  36   0            ! no continuum transfer
switch  37   1            ! do line transfer
switch  38   1            ! symmetric line profiles
switch  52  -1            ! do Stark broadening for everything
switch  53   1            ! include all rays in spectrum
switch  54   0            ! include continuum in spectrum
switch  55   1            ! do continuum lowering
switch  72   1            ! do spectral calculation every n timesteps

param   41  1.0e-10       ! initial timestep
param   44  1.0e-10       ! minimum timestep
param   45  1.0e-8        ! maximum timestep
param   46  1.5           ! maximum timestep fractional increase
param   61  1.e-4         ! frac. population for inclusion of iso-sequence


c -------------------------------------------------------------
c   Edits
c -------------------------------------------------------------

dump all

plot "Bare, H-like POPULATIONS vs R"
  xvar  r
  yvar  y1   1 0 0
  yvar  y    1 0 1 1:4

plot "He-like, Li-like POPULATIONS vs R"
  xvar  r
  yvar  y    1 0 2 1:4
  yvar  y1   1 0 3

plot "TAULINE vs EVLINE"
  xvar  evline   1
  yvar  tauline  1
  yvar  tauline  2

plot "LKAP vs EVLINE"
  xvar  evline   1
  yvar  lkap     1 1
  yvar  lkap     2 1

plot "LKAP vs R"
  xvar  r
  yvar  lkap     1 0 -1
  yvar  lkap     2 0 -1

plot "UVOIGT vs R"
```

```
  xvar  r
  yvar  uvoigt   1
  yvar  uvoigt   2

plot "NRB vs R"
  xvar  r
  yvar  nrb      1
  yvar  nrb      2

plot "JLINE vs R"
  xvar  r
  yvar  jline    1 0 -1

plot "JLINE vs EVLINE"
  xvar  evline   1
  yvar  jline    1 ir1
  yvar  jline    1 ir2
  yvar  jline    1 ir3
  yvar  jline    1 ir4
  yvar  jline    1 ir5

plot "JLINE vs R"
  xvar  r
  yvar  jline    2 0 -1

plot "JLINE vs EVLINE"
  xvar  evline   2
  yvar  jline    2 ir1
  yvar  jline    2 ir2
  yvar  jline    2 ir3
  yvar  jline    2 ir4
  yvar  jline    2 ir5

plot "JLINE vs R"
  xvar  r
  yvar  jline    2 0 -1

plot "NE, TEV, TIV, VEL, ZBAR, RHO vs R"
  xvar  r
  yvar  ne
  yvar  tev
  yvar  tiv
  yvar  vel
  yvar  zbar
  yvar  rho

plot "SPECTRUM vs ENERGY"
  xvar sp_energy
  yvar jsparea    0 ir5

plot "SPECTRUM vs ENERGY"
  xvar sp_energy 10
  yvar jsparea   10 ir5

plot "SPECTRUM vs ENERGY"
  xvar sp_energy 20
  yvar jsparea   20 ir5

plot "SPECTRUM vs ENERGY"
  xvar sp_energy 30
  yvar jsparea   30 ir5

plot "SPECTRUM vs FREQUENCY"
  xvar sp_freq   20
  yvar jsparea   20 ir5

plot "TIME-INTEGRATED SPECTRUM vs FREQUENCY"
```

```
time-integrated snapshot
  xvar sp_freq   20
  yvar jsparea   20 ir5

c
c -- time edits --
c

plot "RHO, NE, TEV, TIV, NION vs TIME"
  xvar  time
  yvar  rho      0 ir1
  yvar  ne       0 ir1
  yvar  tev      0 ir1
  yvar  tiv      0 ir1
  yvar  nion     0 ir1

plot "TAULINE vs TIME"
  xvar  time
  yvar  tauline  1 0 -1
  yvar  tauline  2 0 -1

plot "NRB vs TIME"
  xvar  time
  yvar  nrb      1 ir1
  yvar  nrb      2 ir1

plot "Y1 vs TIME"
  xvar  time
  yvar  y1       1 ir1 0:5

plot "TOTAL EMITTED ENERGY vs TIME"
sp-integrated
  xvar  time
  yvar  jsparea  20 ir5

plot "TOTAL INTEGRATED ENERGY vs TIME"
time-integrated
sp-integrated
  xvar  time
  yvar  jsparea  20 ir5
```

2-d line transfer

```
c              **** overlapping line transfer test ****

c  "An Ne-Like Fe Laser Resonantly Photo-Pumped by Ne X Ly-Alpha Radiation"
c   J. Nilsen, JQSRT 46, 547-556 (1991)


c ------------------------------------------------------------
c   Aliases
c ------------------------------------------------------------

alias n1  1
alias n2  5
alias n3 45

alias n4   6
alias n5  10
alias n6  54
alias n7 100

alias k1   1
alias k2   5
alias k3   6
alias k4  10

alias l1   1
alias l2   3
alias l3   5


c ------------------------------------------------------------
c   Materials
c ------------------------------------------------------------

atoms dl2lu1.dat 10. 20.179
atoms dl7lu1.dat 26. 55.847

regionkl  1 5  1 5   95.
  background  5.26e21 2.104e22 4.0 25.5 7.5
  level 2 10 1 2.014e18

regionkl  6 10 1 5  271.
  background  4.94e21 1.976e22 4.0 25.5 7.5
  level 1  1 1 8.0995e19

regionkl  1 5 6 10  271.
  background  4.94e21 1.976e22 4.0 25.5 7.5
  level 1  1 1 8.0995e19

regionkl  6 10 6 10 271.
  background  4.94e21 1.976e22 4.0 25.5 7.5
  level 1  1 1 8.0995e19


c ------------------------------------------------------------
c   Geometry
c ------------------------------------------------------------

geometry xy

quad  2  5  2  5  0.00 0.01  0.00 0.01  1. 1.
quad  2  5  6  9  0.00 0.01  0.01 0.05  1. 1.
```

```
quad  6  9  2  5  0.01 0.05  0.00 0.01  1. 1.
quad  6  9  6  9  0.01 0.05  0.01 0.05  1. 1.


c -------------------------------------------------------------
c   Radiation
c -------------------------------------------------------------

angles 3

ebins 11 50. 8.e4

linedefault prd

line 1  1  1 1  1 2
  lbins 25  5. 1.05

line 2  2  10 1  10 2
  lbins 25  1. 1.02

line 3  2  10 1  10 6
  lbins 25  5. 1.05

resonance  1 3  0.0

r2file pumpr2a generate

aprd 0.000 0.004
aprd 0.008 0.010
aprd 0.020 0.022
aprd 0.038 0.040


c -------------------------------------------------------------
c   Controls
c -------------------------------------------------------------

tstart 0.
tquit  4.0e-09

restart


c -------------------------------------------------------------
c   Switches and Parameters
c -------------------------------------------------------------

switch  11  1          ! make .plt file
switch  25  0          ! steady-state
switch  28  2          ! don't do any initialization
switch  29  1          ! use constant timesteps
switch  30  10            ! dump every n timesteps
switch  34  11            ! symmetric about k=1 & l=1
switch  35  1          ! don't do bremsstrahlung
switch  36  0          ! don't do continuum transfer
switch  37  1          ! do line transfer
switch  38  1          ! use symmetric line profiles
switch  42  1          ! zone centered mesh

param   41  5.e-10        ! timestep


c -------------------------------------------------------------
c   Edits
c -------------------------------------------------------------

plot "JBAR vs R"
```

```
  xvar  r
  yvar  jbar   1:3

plot "JBAR (line 1) vs KR"
  xvar  kr     0 l1
  yvar  jbar   1 l1
  yvar  jbar   1 l2
  yvar  jbar   1 l3

plot "JBAR (line 2) vs KR"
  xvar  kr     0 l1
  yvar  jbar   2 l1
  yvar  jbar   2 l2
  yvar  jbar   2 l3

plot "JBAR (line 3) vs KR"
  xvar  kr     0 l1
  yvar  jbar   3 l1
  yvar  jbar   3 l2
  yvar  jbar   3 l3

plot "JBAR (line 1) vs LR"
  xvar  lr     0 k1
  yvar  jbar   1 k1
  yvar  jbar   1 k2
  yvar  jbar   1 k3
  yvar  jbar   1 k4

plot "JBAR (line 2) vs LR"
  xvar  lr     0 k1
  yvar  jbar   2 k1
  yvar  jbar   2 k2
  yvar  jbar   2 k3
  yvar  jbar   2 k4

plot "JBAR (line 3) vs LR"
  xvar  lr     0 k1
  yvar  jbar   3 k1
  yvar  jbar   3 k2
  yvar  jbar   3 k3
  yvar  jbar   3 k4

plot "JBAR (line 1) vs TIME"
  xvar  time
  yvar  jbar   1 n4
  yvar  jbar   1 n5
  yvar  jbar   1 n6
  yvar  jbar   1 n7

plot "JBAR (line 2) vs TIME"
  xvar  time
  yvar  jbar   2 n1
  yvar  jbar   2 n2
  yvar  jbar   2 n3

plot "JBAR (line 3) vs TIME"
  xvar  time
  yvar  jbar   3 n1
  yvar  jbar   3 n2
  yvar  jbar   3 n3

plot "LINEITER vs TIME"
  xvar  time
  yvar  lineiter 1:3
```

3-d radiation transfer

```
c    **** steady-state spherical continuum-lowering spectrum calculation ****

c -------------------------------------------------------------
c   Aliases
c -------------------------------------------------------------

alias k0    1
alias k1   13

alias k2    5
alias kmid  7
alias k3    9

alias l0    1
alias lmid  4
alias l1    7

alias m0    1
alias m1    5

alias ir1  kmid
alias ir2  kmid + (l1    - 1) * k1
alias ir3  kmid + (lmid  - 1) * k1
alias ir4  ir2  + (m1 - 1) * k1 * l1

alias is1  k1
alias is2  k1 * l1
alias is3  k1 * lmid
alias is4  k1 * l1 * m1


c -------------------------------------------------------------
c   Materials
c -------------------------------------------------------------

atoms rtdat18.dat 18. 40.
atoms dnl01s0      1.  1.

regionklm  k0 k1 l0 l1 m0 m1 1000.
  element 2 6.11e22

regionklm  k2 k3 l0 l1 m0 m1 1000.
  element 1 6.11e22  1 5


c -------------------------------------------------------------
c   Geometry
c -------------------------------------------------------------

geometry xyz

cone  k0 k1 l0 l1 m0 m1   0.000 0.005  90.0 -90.0  90.0 0.0  1.0 1.0 1.0

c -------------------------------------------------------------
c   Radiation
c -------------------------------------------------------------

angles 4

ebins 100 5. 10000. 1.05

spectrum 40     5.  1000.  1.
```

```
spectrum 20  1000.  2800.  1.
spectrum 40  2800.  4200.  1.
spectrum 20  4200. 10000.  1.


line 1 1 1 1 1 2
  lbins 15 50.0 1.1

line 2 1 1 1 1 3
  lbins 15 15.0 1.1


c -------------------------------------------------------------
c   Controls
c -------------------------------------------------------------

tstart 0.
tquit  4.0e-10

restart


c -------------------------------------------------------------
c   Switches and Parameters
c -------------------------------------------------------------

switch   8   0            ! Carlson angles
switch  11   1            ! make .plt file
switch  29   2            ! use variable timesteps
switch  30  10            ! dump every n timesteps
switch  31   0            ! no temperature calculation
switch  34 133            ! symmetric about k=1, l=1, l=lmax, m=1, m=mmax
switch  36   0            ! no continuum transfer
switch  37   1            ! do line transfer
switch  38   1            ! symmetric line profiles
switch  52  -1            ! do Stark broadening for everything
switch  53   1            ! include all rays in spectrum
switch  54   0            ! include continuum in spectrum
switch  55   1            ! do continuum lowering
switch  72   1            ! do spectral calculation every n timesteps

param   41  1.0e-10       ! initial timestep
param   44  1.0e-10       ! minimum timestep
param   45  1.0e-8        ! maximum timestep
param   46  1.5           ! maximum timestep fractional increase
param   61  1.e-4         ! frac. population for inclusion of iso-sequence


c -------------------------------------------------------------
c   Edits
c -------------------------------------------------------------

dump mesh temperatures kappa_sp emis_sp jbar

editray 1  0.01  1.570 0
editray 2  0.01 -1.570 0
editray 3  0.01  0.001 0
editray 4  1.00  0.001 0

plot "Bare, H-like POPULATIONS vs R"
  xvar  r
  yvar  y1   1 0 0
  yvar  y    1 0 1 1:4

plot "He-like, Li-like POPULATIONS vs R"
  xvar  r
  yvar  y    1 0 2 1:4
```

```
  yvar  y1   1 0 3

plot "TAULINEK vs EVLINE"
  xvar  evline   1
  yvar  taulinek 1 1
  yvar  taulinek 1 4
  yvar  taulinek 2 1
  yvar  taulinek 2 4

plot "TAULINEL vs EVLINE"
  xvar  evline   1
  yvar  taulinel 1 1
  yvar  taulinel 1 7
  yvar  taulinel 2 1
  yvar  taulinel 2 7

plot "LKAP vs EVLINE"
  xvar  evline   1
  yvar  lkap     1 1
  yvar  lkap     2 1

plot "LKAP vs R"
  xvar  r
  yvar  lkap     1 0 -1
  yvar  lkap     2 0 -1

plot "UVOIGT vs R"
  xvar  r
  yvar  uvoigt   1
  yvar  uvoigt   2

plot "NRB vs R"
  xvar  r
  yvar  nrb      1
  yvar  nrb      2

plot "JLINE vs EVLINE"
  xvar  evline   1
  yvar  jline    1 ir1
  yvar  jline    1 ir2
  yvar  jline    1 ir3
  yvar  jline    1 ir4

plot "JLINE vs R"
  xvar  r
  yvar  jline    2 0 -1

plot "JLINE vs EVLINE"
  xvar  evline   2
  yvar  jline    2 ir1
  yvar  jline    2 ir2
  yvar  jline    2 ir3
  yvar  jline    2 ir4

plot "JLINE vs R"
  xvar  r
  yvar  jline    2 0 -1

plot "NE, TEV, TIV, VEL, ZBAR, RHO vs R"
  xvar  r
  yvar  ne
  yvar  tev
  yvar  tiv
  yvar  vel
  yvar  zbar
  yvar  rho
```

```
plot "SPECTRUM vs R"
  xvar r
  yvar jsp        0  0  20
  yvar jsp        0  0  65
  yvar jsp        0  0  70
  yvar jsp        0  0  75
  yvar jsp        0  0 100

plot "SPECTRUM vs ENERGY"
  xvar sp_energy
  yvar jsp        0 ir1
  yvar jsp        0 ir2
  yvar jsp        0 ir3
  yvar jsp        0 ir4

plot "ISP vs ENERGY"
send-to-dump
  xvar sp_energy
  yvar isp+       0 is1 0 1
  yvar isp+       0 is2 0 2
  yvar isp+       0 is3 0 3
  yvar isp+       0 is4 0 4

plot "ISP vs YZ"
slice k k1
send-to-dump
  xvar y3d
  xvar z3d
  yvar isp+       0  0 75 1
  yvar isp+       0  0 75 2

plot "ISP vs XZ"
slice k k1
send-to-dump
  xvar x3d
  xvar z3d
  yvar isp+       0  0 75 3

plot "ISP vs XY"
slice k k1
send-to-dump
  xvar x3d
  xvar y3d
  yvar isp+       0  0 75 4

plot "ISP vs R"
send-to-dump
  xvar r
  yvar isp+       0  0 75 1
  yvar isp+       0  0 75 2
  yvar isp+       0  0 75 3
  yvar isp+       0  0 75 4

c
c -- time edits --
c

plot "RHO, NE, TEV, TIV, NION vs TIME"
  xvar  time
  yvar  rho      0 ir1
  yvar  ne       0 ir1
  yvar  tev      0 ir1
  yvar  tiv      0 ir1
  yvar  nion     0 ir1

plot "NRB vs TIME"
  xvar  time
```

```
  yvar  nrb      1 ir1
  yvar  nrb      2 ir1

plot "Y1 vs TIME"
  xvar  time
  yvar  y1       1 ir1 0
  yvar  y1       1 ir1 1
  yvar  y1       1 ir1 2
  yvar  y1       1 ir1 3
  yvar  y1       1 ir1 4
  yvar  y1       1 ir1 5

plot "JBAR vs TIME"
  xvar  time
  yvar  jbar     1 ir1
  yvar  jbar     1 ir2
  yvar  jbar     1 ir3
  yvar  jbar     1 ir4

plot "JBAR vs TIME"
  xvar  time
  yvar  jbar     2 ir1
  yvar  jbar     2 ir2
  yvar  jbar     2 ir3
  yvar  jbar     2 ir4

plot "LINEITER vs TIME"
  xvar  time
  yvar  lineiter 1
  yvar  lineiter 2
```

Index of Commands and Edits

—E—

—F—

—G—

—H—

—I—

—J—

—K—

—L—

—M—

—N—

—P—

—Q—

—R—

—S—

—T—

—U—

—V—

—W—

—X—

—Y—

—Z—

Appendix: ULTRA and PDBVIEW basics

ULTRA is a program for the presentation, manipulation and analysis of 1-d data sets, i.e. curves. It can read and write ASCII data files or PDB files with ULTRA curves in them, such as the *.ult* files produced by CRETIN.

Before using ULTRA on a Unix system, you must define an ULTRA environment variable, specifying the directory for various configuration and extension files. The default behavior of ULTRA may be modified with a *.ultrarc* file. The following entries in *.ultrarc* are recommended for use with CRETIN (Note: the parentheses are required and the comments must be placed on lines beginning with semicolons):

```
(label-length 60)          [allow viewing the entire curve label created by CRETIN]
(label-space 0.2)          [allow space for labels under the graphic display]
(label-type-size 10)       [font size for labels]
```

In ULTRA commands, spaces delimit items on the input line. Semicolons may be used to stack multiple commands on a single input line. Ranges of curve numbers or letters may be indicated using colon notation, e.g. 3:6 or a:d.

To start ULTRA and open file *data.ult*, type
ultra *data.ult*
or to open file *data.ult* from within ULTRA, type
rd *data.ult*
More than one file may be open simultaneously.

To list the curves in *data.ult*, type
menu
or to list all curves with labels starting with *abc*, type
menu *abc***
or to list all curves with labels containing the string *abc*, type
menu "* *abc* ***"**
or to send the list of curves to a file, type
print-menu *[filename]*
where *filename* is optional and defaults to "ultra.menu".

To display curves numbered *8* and *12*, type
cur *8 12*

To list the curves currently displayed, type
lst

To delete curves labeled *e* and *g* (where the labels are those on screen or returned by **lst**), type
del *e g*

To erase all curves currently displayed, type
era

To send the current graphics display to a postscript file (defaulted to "plots.ps"), type
print

To exit ULTRA, type
end

ULTRA has many capabilities for manipulating and combining curves. Descriptions of most commands can be obtained from ULTRA's help capabilities. To list available commands, type

help

To obtain information about a given command, type

help *command*

If you don't know the name of a command, choose an appropriate keyword and type

apropos *keyword*

PDBVIEW is a program for working with PDB files, such as the various output files (*.ult, .rxx, .sxx, .dxx*) produced by CRETIN. It allows browsing PDB files graphically or textually and also allows editing PDB files.

Before using PDBVIEW, you must define a SCHEME environment variable, specifying the directory for various configuration and extension files.

When dealing with a *.ult* file, PDBVIEW can graphically display both 1-d data sets (curves) and 2-d data sets (mappings). To list all curves and mappings, type

menu

or to list only mappings, type

menu "*%*"

(This takes advantage of the titles assigned to 2-d and 3-d data sets by CRETIN).

To display curve/mapping number *5*, type

dm *5*

The default rendering for a 2-d mapping is a contour plot. To change the rendering, type

vr *mode*

where *mode* is one of the following: "contour", "image", "fill-poly", "shaded", "wire-frame", or "mesh".

To list the curves/mappings currently displayed, type

lsv

To delete the mapping labeled *2* (where the label is that returned by **lsv**), type

dl *2*

To delete all mappings from the window, type

clv

To send the current graphics display to a postscript file, type

hc [color]

where **color** must be specified on the first **hc** command for a color postscript file.

PDBVIEW can display multiple windows simultaneously, and multiple viewports in a given window. See the PDBVIEW documentation or help package for more information. The above commands are appropriate for the current window with a single viewport.

To create a new window, type

cw "*title***"**

where *title* will appear at the top of the new window. The quotes are necessary if *title* is ascii, but not if *title* if numeric. A **cw** command with no arguments will cycle through the existing windows, making each one active in turn.

When dealing with a general PDB file, such as a dump file (*.rxx*, *.sxx*, *.dxx*) from CRETIN, PDBVIEW can list and edit data from the file. To list the contents of a file, type

ls

The file may contain a directory structure which may be navigated with Unix commands.

To describe the structure and format of variable *var*, type

desc *var*

To list the data contained in variable *var*, type

print *var*

or simply

var

To list the subset (*i*=2-4,*j*=7-10) of a 2-d array *var(i,j)*, type

[**print**] *var[2:4,7:10]*

To change the dimensions of a variable (for the current PDBVIEW session only), type

chdim *var dim1 dim2 dim3* …

To set element 9 of a 1-d array *var(i)* to have *value*, type

change *var[9] value*

or simply

var[9] value

To plot a variable *yvar* against a 1-d variable *xvar*, type

plot *yvar xvar*

To plot a variable *zvar* on a 2-d mesh consisting variables *xvar* and *yvar*, type

plot *zvar ''(xvar yvar)*

To exit PDBVIEW, type

end

PDBVIEW has many more capabilities for viewing and manipulating data. To list the available commands, type

help

or to receive a description of *command*, type

help *command*

ULTRA, PDBVIEW and PDB are part of Portable Application Code Toolkit (PACT).